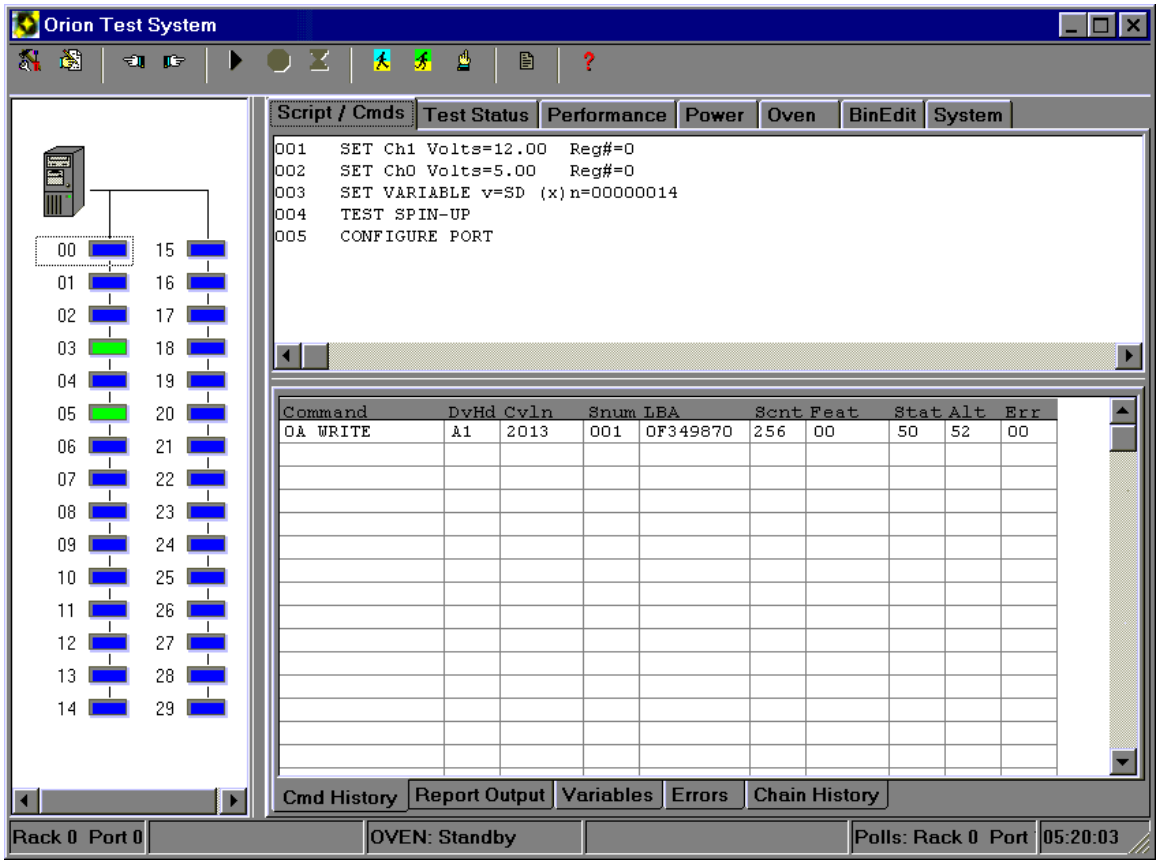
FLEXSTAR TECHNOLOGY

Orion Host Software

Users Manual



Document # 98-36391-00 Rev 1.23 09/18/2003

1.0 Table of Contents

1.0 Table of Contents *2*

Table of Figures *7*

2.0 Read Me First *8*

2.1 Orion Host Software Packing List *8*

2.2 System Requirements *8*

3.0 Introduction *9*

3.1 Orion Host Software Description *9*

3.2 Product Features *9*

4.0 Installation *11*

4.1 Remote Boot Software Installation *11*

4.2 Orion Software Installation *11*

5.0 Orion Host Software Operating Instructions *12*

5.1 System Setup *12*

5.1.1 Server Options *13*

5.1.2 Client Options *14*

5.1.3 Directories *17*

5.1.4 Output Options *18*

5.1.5 Oven Options *21*

5.1.6 Preferences *24*

5.1.7 Bootup Options *25*

5.2 OPERATION *25*

5.2.1 Port Icon Definitions *25*

5.2.2 Resetting Ports *27*

5.2.3 Starting a Test *27*

5.2.4 Stopping a Test *28*

5.2.5 Pausing and Resuming a Test *28*

5.3 ORION HOST GUI *29*

5.3.1 Script / Commands Page *29*

5.3.1.1 Command History Page *30*

5.3.1.2 Report Output Page *30*
5.3.1.3 Variables Page *31*
5.3.1.4 Errors Page *32*
5.3.1.5 Chain History Page *33*
5.3.2 Test Status Page *34*
5.3.3 Performance Page *35*
5.3.4 Power Display Page *36*
5.3.5 Oven Status Page *37*
5.3.6 Binary Editor Page *38*
5.3.7 System Page *39*
5.3.8 Toolbar Buttons *40*
5.3.8.1 Options Button *41*
5.3.8.2 Editors Button *41*
5.3.8.3 Previous Rack Button *41*
5.3.8.4 Next Rack Button *42*
5.3.8.5 Start Oven Button *42*
5.3.8.6 Stop Oven Button *42*
5.3.8.7 Oven Pause/Resume Button *42*
5.3.8.8 Macro Step Mode Button *42*
5.3.8.9 Command Step Mode Button *42*
5.3.8.10 Advance Step Button *43*
5.3.8.11 Generate Report Button *43*
5.3.8.12 Help Button *43*
5.4 UTILITIES *44*
5.4.1 Test Script Editor *44*
5.4.2 Oven Script Editor *44*
5.4.3 Binary Editor *45*
5.4.3.1 Opening a File *45*
5.4.3.2 Saving Editor Contents to a File *45*
5.4.3.3 Loading from Client Buffer *46*
5.4.3.4 Saving to a Client Buffer *46*
5.4.3.5 Printing the Editor Contents *46*
5.4.3.6 Buffer Selection *46*

5.4.3.7 Other Features *46*
5.5 DATABASE OUTPUT *47*
5.5.1 Legacy Files *47*
6.0 Orion Script Language *48*
6.1 USING THE SCRIPT EDITOR *48*
6.1.1 Starting the Editor *48*
6.1.2 Creating a New Script *48*
6.1.3 Inserting New Script Commands *49*
6.1.4 Modifying Parameters *49*
6.1.5 Deleting Script Commands *49*
6.1.6 The Replace Command *49*
6.1.7 Save Comments in Script *50*
6.1.8 Saving Script Programs *50*
6.1.9 Printing Script Programs *50*
6.1.10 Editing Abort Limits *50*
6.1.11 Copy, Cut, and Paste *50*
6.1.12 Enhanced Script Feature *50*
6.2 SCRIPT LANGUAGE DESCRIPTION *51*
6.2.1 COMMANDS BY GROUP *51*
6.2.1.1 ATAPI Commands *51*
6.2.1.2 Loop & Utility Commands *56*
6.2.1.3 Oven Control Commands *63*
6.2.1.4 IDE Commands *64*
6.2.1.5 Program Control Commands *77*
6.2.1.6 Read Commands *91*
6.2.1.7 Write Commands *95*
6.2.1.8 SCSI Commands *98*
6.2.1.9 CCS Command Sub-Group *107*
6.2.1.10 Command Terminators *118*
6.2.1.11 Set Parameter Commands *118*
6.2.1.12 Seek Commands *124*
6.2.1.13 Test Macro Commands *128*
6.2.1.14 Voltage Margin Commands *133*

6.3 VARIABLES *137*
6.3.1 VARIABLE DETAILS *140*
6.4 ABORT LIMITS *154*
7.0 Report Generator Description *159*
7.1 Report Generator Script Language Description *159*
7.1.1 Orion Report Generator Command List *160*
7.1.2 Keyword List *166*
8.0 Oven Control Description *168*
8.1 Oven Script Editor Operation *169*
8.1.1 Opening an Existing Script File *169*
8.1.2 Saving a Script File *169*
8.1.3 Creating a New Script *170*
8.1.4 Adding New Commands *170*
8.1.5 Deleting Commands *170*
8.1.6 Replacing Commands *170*
8.1.7 Checking Program Syntax *170*
8.2 Oven Script Language Description *171*
8.2.1 Autostart *171*
8.2.2 Comment *171*
8.2.3 Events *171*
8.2.4 LoopA, LoopB, LoopC *171*
8.2.5 Ramp *171*
8.2.6 Soak *171*
8.3 Oven Synchronization *172*
8.3.1 Oven Slave Mode *172*
8.3.2 Oven Master Mode *174*
9.0 Troubleshooting *176*
9.1 Server will not communicate with clients *176*
10.0 Technical Support *177*
11.0 Glossary *178*
12.0 Appendix A - Image File Format *180*
13.0 Appendix B – Interpreting the Test Report *188*
14.0 Appendix C – Client Console Function *196*

15.0 Appendix D - Data Compare Description 197
16.0 Appendix E – Command Timeout Description 202
17.0 Appendix F – On Error Gosub Description 203
18.0 Appendix G – Soft/Hard Error Description 204
19.0 Appendix H – Error Rate Calculations 205
20.0 Appendix I – Large File Download Example 206

Table of Figures

Figure 1 - Enter System Setup 12
Figure 2 - Server Options 13
Figure 3 - Client Options 15
Figure 4 - Client Properties Table 16
Figure 5 - Directories 18
Figure 6 - Output Options 19
Figure 7 - Oven Options 21
Figure 8 - Preferences 24
Figure 9 - Port Icon Definitions 26
Figure 10 - Port Group Selection 27
Figure 11 - Script / Commands Page 29
Figure 12 - Report Output Page 30
Figure 13 - Variables Page 31
Figure 14 - Errors Page 32
Figure 15 - Chain History Page 33
Figure 16 - Test Status Page 34
Figure 17 - Performance Page 35
Figure 18 - Power Display Page 36
Figure 19 - Oven Status Page 37
Figure 20 - Binary Editor Page 38
Figure 21 - Systems Page 39
Figure 22 - Toolbar Buttons 40
Figure 23 - Binary Editor Description 45
Figure 24 - Script Editor 48
Figure 25 - Report Generator Script Editor 160
Figure 26 - Oven Script Editor 169
Figure 27- Oven Script Example 172
Figure 28 - Oven Slave Example 173
Figure 29 - Oven Master Example 174
Figure 30 - Interpreting the Test Report 190

2.0 Read Me First

2.1 Orion Host Software Packing List

The Orion Host software package includes the following items:

- CD-ROM disk containing all server side installation files.
- Orion Host Users Manual (this document) P/N 98-36391-00.
- Orion FBoot Software User Manual P/N 98-35783-00.

2.2 System Requirements

The Orion Host Software requires the following minimum system attributes:

- IBM compatible PC.
- 128 MB of system RAM.
- 500 MHz Pentium CPU.
- 30 MB of hard disk space available.
- 8x CD-ROM drive.
- Ethernet 10/100 base-T adapter.
- Mouse.
- Super VGA video adapter capable of at least 1024x768 resolution.

3.0 Introduction

3.1 Orion Host Software Description

The Orion Host software is a Windows 98 / NT 4.0 / 2000 compatible application, which provides control and data collection for all *clients* connected to the central *server* system. The application presents a graphical view of all connected clients and detailed activity of a selected test node.

Each *client* (SBC) in the system can have from one to eight test nodes (devices under test) associated with it. Test nodes are represented as device icons on the HOST port view screen and are color coded to show various status conditions.

Device tests are performed by writing and executing test program scripts, which are edited using the Script Editor supplied with the software.

The Orion software adds many new features to enhance the operation and data collection capabilities.

The Orion Host software is designed to be compatible with the Flexstar DOS HOST software used in earlier test products however some differences do exist:

- Some performance measurements are different due to the difference in CPU clock speeds and execution times.

- Orion script editor pads labels with spaces. This can cause compatibility problems since the spaces become part of the label string. This problem can be avoided by filling unused spaces in the labels with the underscore character '_'.

The ORION Host software can start, stop, pause, and resume any test. The Host software includes power margining control and graphic performance plotting for real-time analysis of current test data through a single stand-alone GUI. Opening, editing, and saving scripts are fast and effective through the easy to use GUI environment.

3.2 Product Features

The FLEXSTAR Orion Host has the following features:

- Powerful multi-tasking Windows compliant software.
- Integrated temperature and humidity control.
- Capability to network up to 144 clients together.
- FlexStar legacy compatibility.

- Operates with Windows 98, NT 4.0, and 2000 operating systems.

4.0 Installation

4.1 Remote Boot Software Installation

Please refer to the FBoot user manual P/N 98-35783-00. Installing FBoot first will insure the network adapter is setup correctly.

4.2 Orion Software Installation

The Orion software is distributed on CD-ROM. The following sequence is used to install the software:

- Insert the Orion software CD into the CD-ROM drive.
- Select Run… from the Windows START menu.
- Use the Browse button to locate the Host SETUP.EXE file in the Host folder on the CD. Then click OK
- Follow the setup instructions on the screen.
- Following the installation, there will be a file named SUB-FOLDERS.ZIP in the ORION directory. Right click on this file and drag it to the root of the C: drive (‘C:\’). Using WinZip or equivalent, extract this ZIP file to C:\. This will create certain required directories and load sample files within these new directories.

5.0 Orion Host Software Operating Instructions

5.1 System Setup

All Orion Host software options and preferences are contained in a file called O25HOST.INI. This file resides in the application's working directory (usually C:\Program Files\Flexstar\Orion). Note that this INI file is encrypted and cannot be read with a text editor.

The options are set up by clicking the options button on the main window toolbar and selecting ***Options*** on the dropdown menu. A Setup dialog box (shown below) will then appear.

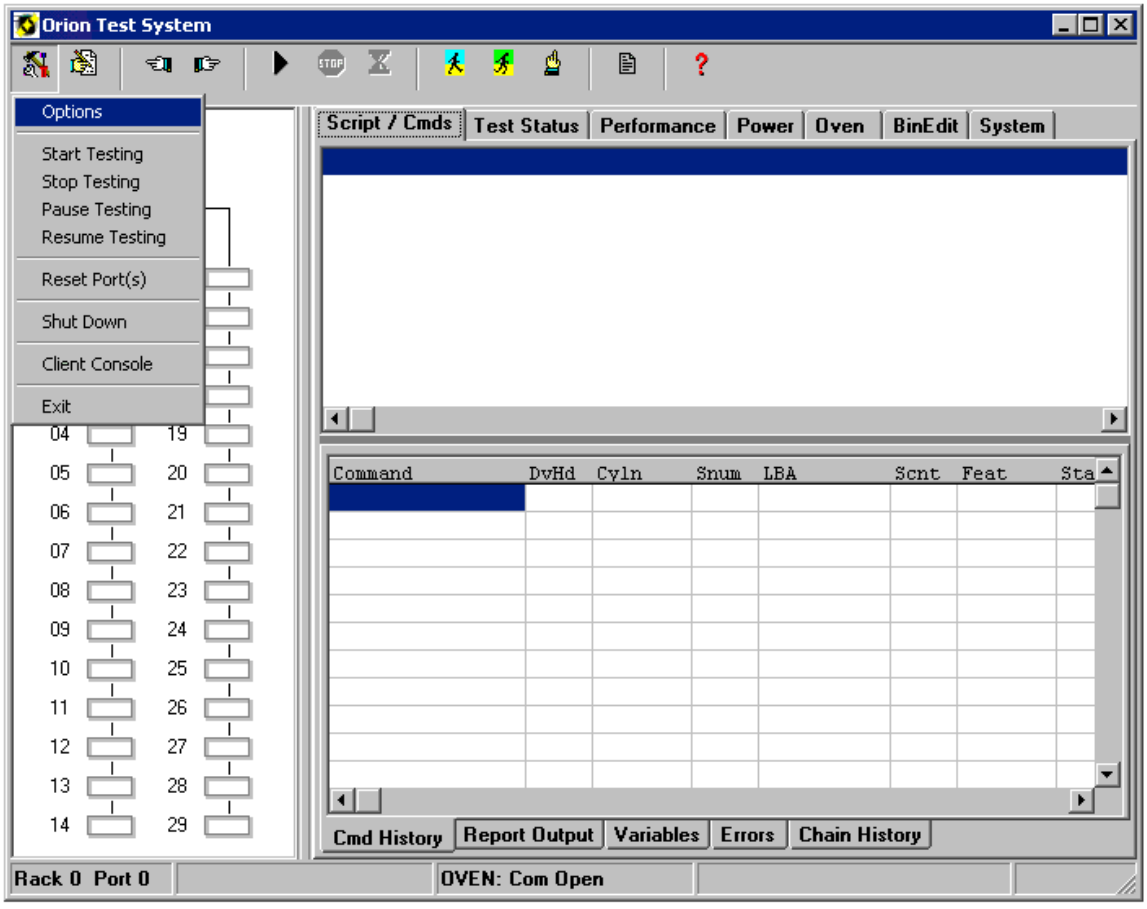


Figure 1 - Enter System Setup

5.1.1 Server Options

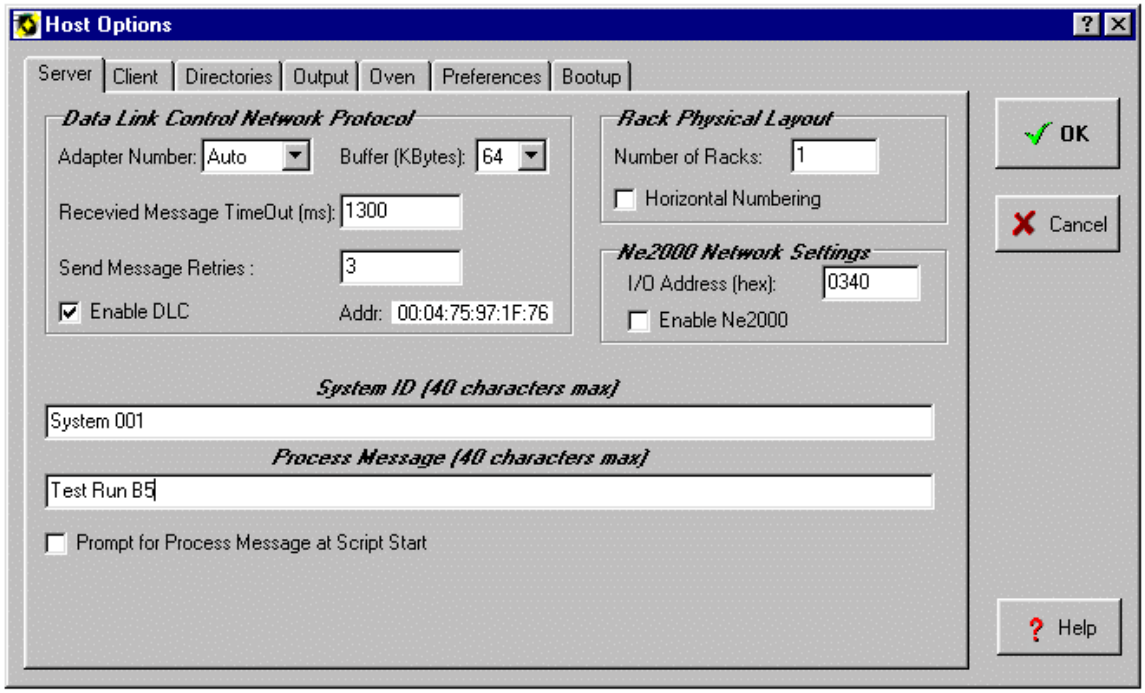


Figure 2 - Server Options

Data Link Control Network Protocol – This section controls the **D**ata **L**ink **C**ontrol (DLC) type of networking.

- ***Adapter Number*** – Defines the system adapter number (Auto, 0, or 1). Default = Auto.
- ***Buffer (Kbytes)*** – Defines the size of the communications buffer. Default = 64K.
- ***Received Message Timeout (ms)*** – Sets the timeout value in milliseconds for expected client messages. Default = 1500 ms.
- ***Send Message Retries*** – Number of retries performed on message timeout or error. Default = 2.
- ***Enable DLC*** – Check this box to enable the DLC communications method.

Rack Physical Layout – This section is used to define the systems physical layout.

- ***Number of Racks*** – Number of separate systems connected to the server. Default = 1.

- ***Horizontal Numbering*** – Numbers ports left to right instead of the default top to bottom.

NE2000 Network Settings – This section defines settings for the NE2000 card used when the Flexstar communications protocol is used.

- ***I/O Address (hex)*** – Defines the base address in I/O space of the NE2000 NIC card. Default = 340 hex.

- ***Enable NE2000*** – Enables the Flexstar communications protocol and the use of the NE2000 card for the private LAN.

System ID – A 16 character user defined string that appears in the ASCII report files and is normally used to identify the actual rack system. This field also appears in the title window.

Process Message – A 16 character user defined string that appears in the ASCII report files under 'Comments'.

5.1.2 Client Options

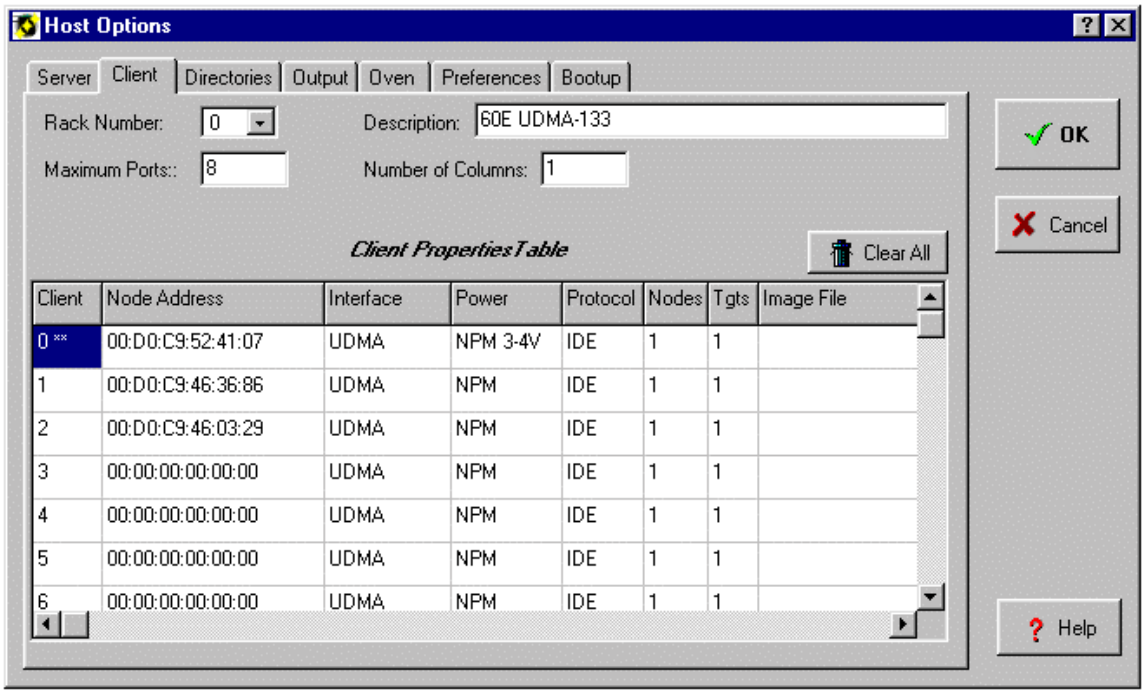


Figure 3 - Client Options

Rack Number – Selects the client properties table for the specified rack. This only applies when more than one rack system is connected to the server.

Maximum Ports – Defines the actual number of test ports (nodes) that are present in the currently selected rack.

Description – This string is a descriptor for each rack attached to the system. This is a visual aid only and is not used anywhere else.

Number of Columns – This entry defines the number of columns that are drawn on the Orion rack view GUI. The normal value is 2 but this can be set to any number which corresponds to the physical number of columns in the system being represented.

Clear All Button – Allows the user to clear the contents of the Client Properties Table. This can be used when building a client table from scratch.

Client Properties Table – This table associates the clients Ethernet address with it's physical position within the rack. This table also defines specific hardware resources that are used by the client. The number of test nodes is also defined here to indicate how many devices can be connected to an individual client.

To edit the table, double click on the specific client cell MAC address in the table. This will bring up the client properties dialog box that allows editing of the Ethernet (MAC) address and the hardware assignments for that client:

Note: These properties are stored in the O25Host.INI file. If this file is missing as a result of a new installation or accidental erasure, the client properties will assume default values. Also note that the O25Host.INI file can be moved to another server to avoid having to retype all entries.

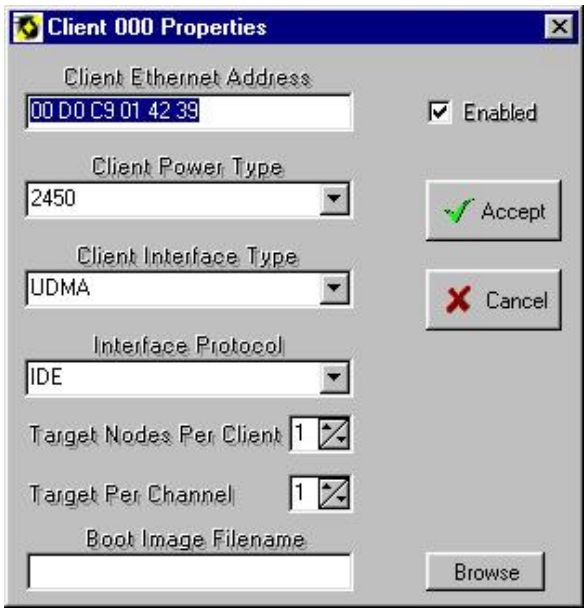


Figure 4 - Client Properties Table

Each sequential entry in the client network table represents a physical test node position. For example, the first table entry position is port 0 in a typical system (upper left corner). As the number increases, the physical position in the rack moves toward the bottom and then wraps to the top of the next column. When the number increases beyond the limit of one rack, this represents the start (upper left corner) of the next rack.

Note that if a physical client in the system is removed; entering all zeros for the Ethernet address for that physical position will disable the client and force the software to skip this position. It is more advisable however, to disable the port rather than change the MAC address. This acts as a placeholder for that client position so it can be easily replaced in the future.

Client Ethernet Address – The client Ethernet address is a unique address that is 'hardwired' into each Ethernet chip. This address is used to associate a port on the network with its physical location in the test system. An Ethernet (MAC) address is a 6-byte number that cannot be duplicated by any other SBC.

Client Power Type – Power type refers to the specific programmable power board used in the Flexstar test system. There are several possible types:

1. 2450 – First generation power board.

2. PC104 – Newer design for use in the PC104 slot of the Orion SBC. ***NOTE***: Select "1 channel margin" to enable PC104.

3. 1, 2, 4 channel margin power cards – older ISA power cards used in ATX motherboards.

4. NPM – Network power margin board used mainly on the Pegasus SBC's.
5. NPM 3-4V – Network power margin board used to supply up to 4 voltages per target.
6. MPS-3 – Third party low cost power card. No longer supported.

Client Interface Type – Identifies the hardware type of the host bus adapter: SCSI, IDE, 1394, etc.

Interface Protocol – Identifies the protocol for the given hardware. This can be important for hardware that can support more than one protocol. Example: IDE can support both ATA and ATAPI protocols.

Target Nodes Per Client – This identifies how many devices can be attached to each client. The Host software will reserve this many slots for each client in the system.

Target Per Channel – This defines how many channels of a given host bus adapter will be attached to devices. For example, a client can have two 2-channel HBA's installed. If both channels of both cards will be used, *Target Nodes Per Client* should = 4 and *Target Per Channel* should = 2.

Boot Image Filename – Not currently used.

5.1.3 Directories

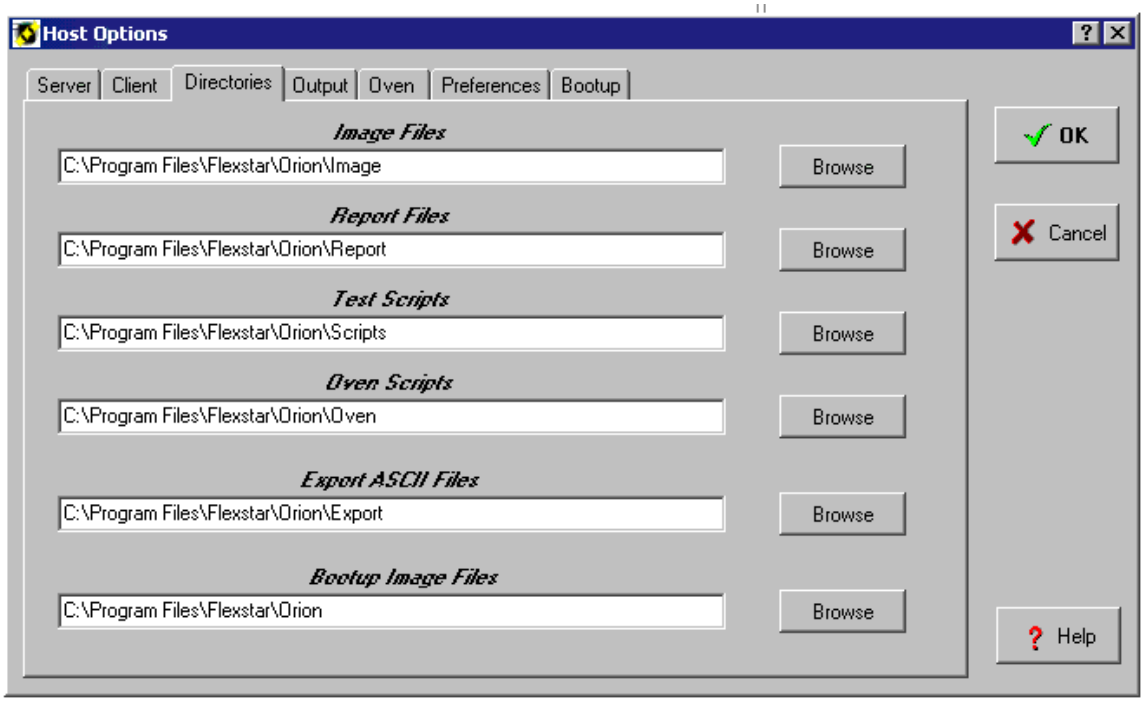


Figure 5 - Directories

The directories edit fields allow the user to assign storage directories for output files, test scripts, etc. The default directory is the main application directory. By clicking the browse button next to each edit box, the user can browse for the appropriate directory.

Image Files - Defines the directory where the software writes the IMAGE database files.

Report Files – Defines the directory where the software writes the REPORT output files.

Test Scripts - Defines the directory where the TEST SCRIPTS are acquired.

Oven Scripts - Defines the directory where OVEN SCRIPTS are acquired.

Export ASCII Files - Defines the destination path for custom ASCII database files.

Bootup Image Files – Not currently used.

5.1.4 Output Options

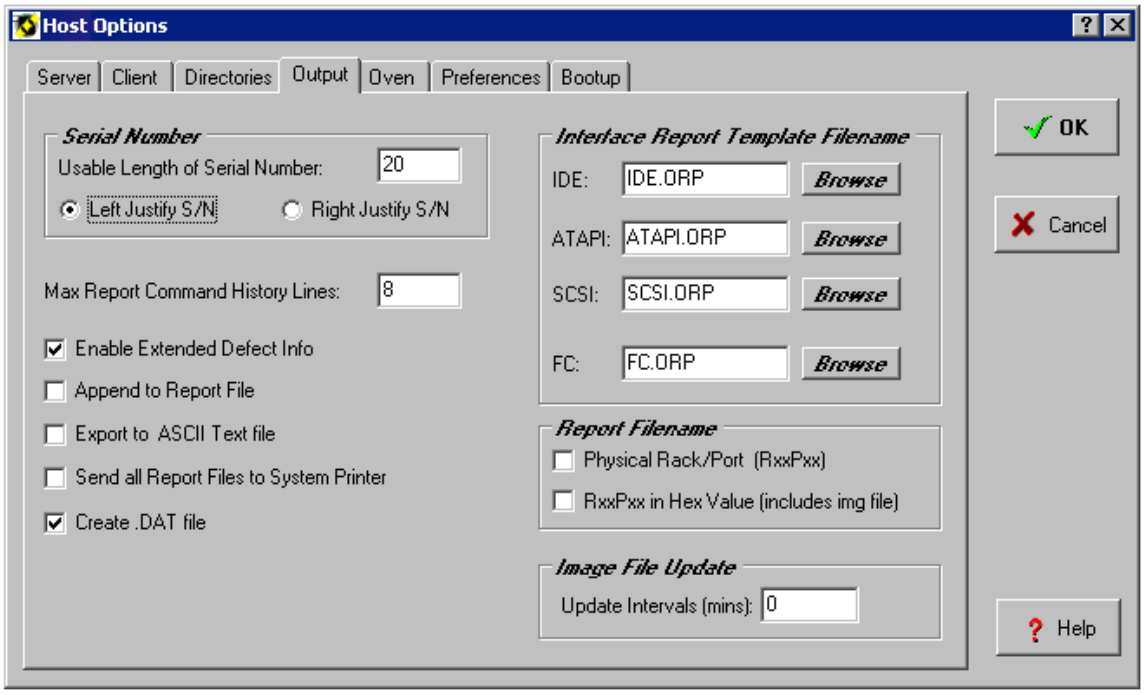


Figure 6 - Output Options

Serial Number – This section defines properties for the device serial number.

- ***Usable Length of Serial number*** – Maximum number of characters used in the serial number string. This field can be from 1 to 20. Default = 20.

- ***Left / Right Justify Serial Number*** – These option selects define whether the serial number string is LEFT or RIGHT justified. If the "Usable Length of Serial number" is set to 20, the option setting is ignored.

Max Report Command History Lines – This field defines the number of command history items that will appear on the final ASCII Report file. The number can be from 0 to 255. Default = 8.

Enable Extended Defect Info – This checkbox allows compatibility with the old style legacy test system, which contained less information on detected errors. This should normally be enabled.

Append to Report File – This checkbox allows the report file to be APPENDED to rather than overwritten. This is useful to maintain a history of reports for a given test port.

Export to ASCII Text File – Currently not supported.

Send all Report Files to System Printer – This option allows the ASCII report file to be sent to

the system printer for hardcopy.

Create .DAT file – This option creates a duplicate of the IMAGE file but names the file using the serial number of the device tested instead of the RxxPxxDx.IMG format.

Interface Report Template Filename – This section sets up specific report generator script files (*.ORP) for various interface types. These script files can be customized to create virtually any type of formatted report output.

- ***IDE*** – Defines the ORP file for IDE (ATA) interfaces - usually IDE.ORP.
- ***ATAPI*** - Defines the ORP file for ATAPI (CD-ROM) interfaces – usually ATAPI.ORP.
- ***SCSI*** - Defines the ORP file for SCSI interfaces – usually SCSI.ORP..
- ***FC*** - Defines the ORP file for Fibre Channel interfaces – usually FC.ORP.

Report Filename – Sets options for the naming of the report output file(s).

- ***Physical Rack/Port*** – The report file (*.REP) is named using the "RxxPxxDx.IMG" format instead of using the device serial number as the file name.
- ***RxxPxx in Hex Value (includes img file)*** – Uses HEX values instead of decimal to represent rack and port numbers. This is necessary when the number of ports controlled by the server exceeds 99.

Image File Update – This section defines properties for automatically updating the image file (refresh).

- ***Update Interval (mins)*** – This field determines how often the database IMAGE files are refreshed in minutes. If the number = 0, the IMAGE files are not automatically updated.

5.1.5 Oven Options

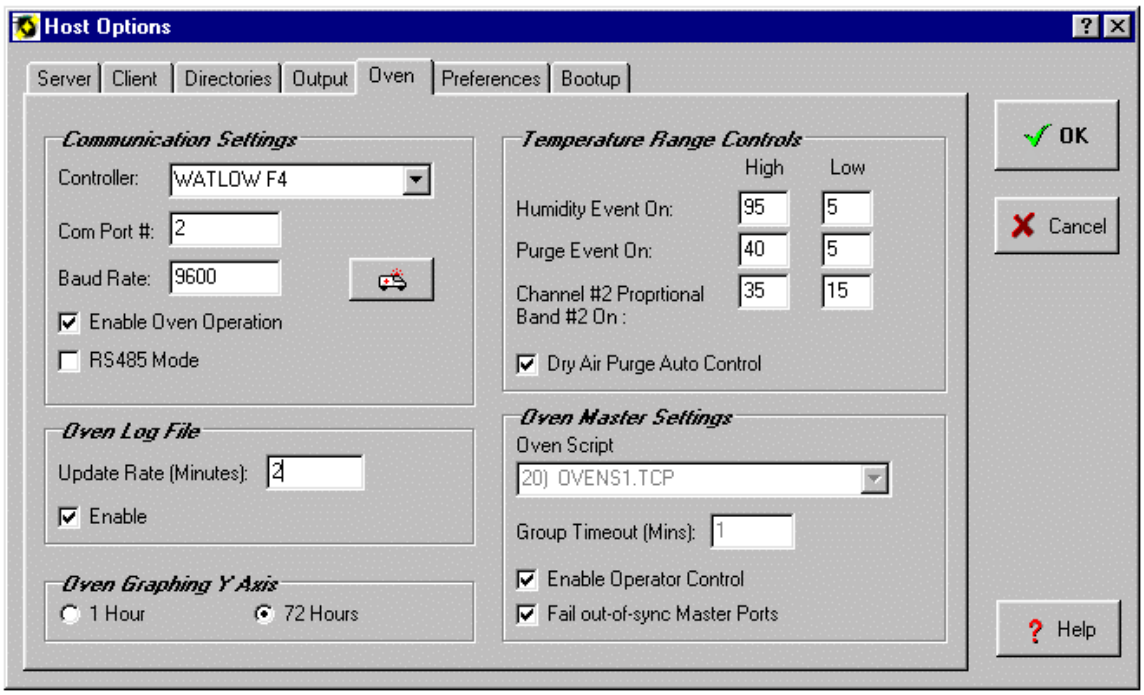


Figure 7 - Oven Options

Communication Settings – This section sets the communications properties for the server RS-232 port which is connected to the oven's temperature controller.

- ***Controller*** – Defines the type of controller that is used in the oven.
- ***Com Port #*** - The number of the server Com port that is connected to the oven controller. Default = 1.
- ***Baud Rate*** – Used to set the RS-232 serial communications speed. Can be set to 9600 or 19200. Default = 19200.
- ***RS-485 Mode*** – This is a special mode used when multiple chamber temperature controllers (up to 4) are connected to one server. RS485 mode is also required when using the Chromalox 1604 controller. The RS-232 is converted to RS-485 through an external adapter. When using this mode, the oven controllers must also be programmed to accept RS-485 interface signals (see Oven Manuals).
- ***Enable Oven Operation*** – This checkbox must be enabled for the system to communicate

with a chamber temperature controller.

Oven Log File – This section sets properties for creating a LOG file to track oven operation. The log file is called OVEN.LOG. Each entry in the log file contains a time stamp, current temperature set point, current temperature, humidity set point, and actual humidity.

- ***Update Rate (Minutes)*** – Determines how often samples are taken and recorded in the log file.

- ***Enable*** – Enables the logging function.

Oven Graphing – This section sets properties for the oven graph.

- ***1 Hour / 72 Hours*** – Sets the graph X axis scale for 1 hour or 72 hours. The Host can store up to 72 hours of oven operation. Beyond this, the graph will continue to scroll showing the last 72 hours of operation.

Temperature Range Controls – This section sets ranges for certain oven EVENTS.

- ***Humidity Event ON*** – Defines the range ***inside*** of which Humidity control is enabled. Outside of this range, humidity will be disabled.

- ***Purge Event ON*** - Defines the range ***inside*** of which Dry Air Purge is enabled. Outside of this range, Dry Air Purge will be disabled.

- ***Channel #2 Proportional Band #2 ON*** – (Used with Chromalox 2030 controller only). Defines the range ***inside*** of which the proportional band #2 is enabled. Outside this range, channel #1 proportional band is used. This function sets different parameters for Humidity control to achieve better performance within the full operational range of the oven.

- ***Dry Air Purge Auto Control***

  This feature, when enabled, will prevent the dry air purge from being activated (disables event 4 of the temperature controller) if the averaged actual humidity is below the set point. This prevents the chamber's frozen coil de-humidifier from freezing up.

Oven Master Settings – Miscellaneous oven properties.

- ***Oven Script*** – This combo-box contains the names of all available oven script files. When a test script starts an oven run, this is the oven script that will be executed.

- ***Group Time-out*** – The group time-out is a watchdog timer that monitors all ports that are attempting to master the oven. If one or more ports within the group do not "register" within the specified time-out period, those ports are removed from the mastering group.

- ***Enable Operator Control*** – Allows the user to manually set the script name in the oven section of the status window.

- ***Fail Out of Sync Master Ports***

  If enabled, any port that falls out of sync when running an oven master group operation will fail

for “OUT OF OVEN SYNC”.

Reset Oven Communications – (Button with the ambulance icon) Re-initializes the RS232C port settings. This is useful if the baud rate has been changed either on the server end or the temperature controller end.

5.1.6 Preferences

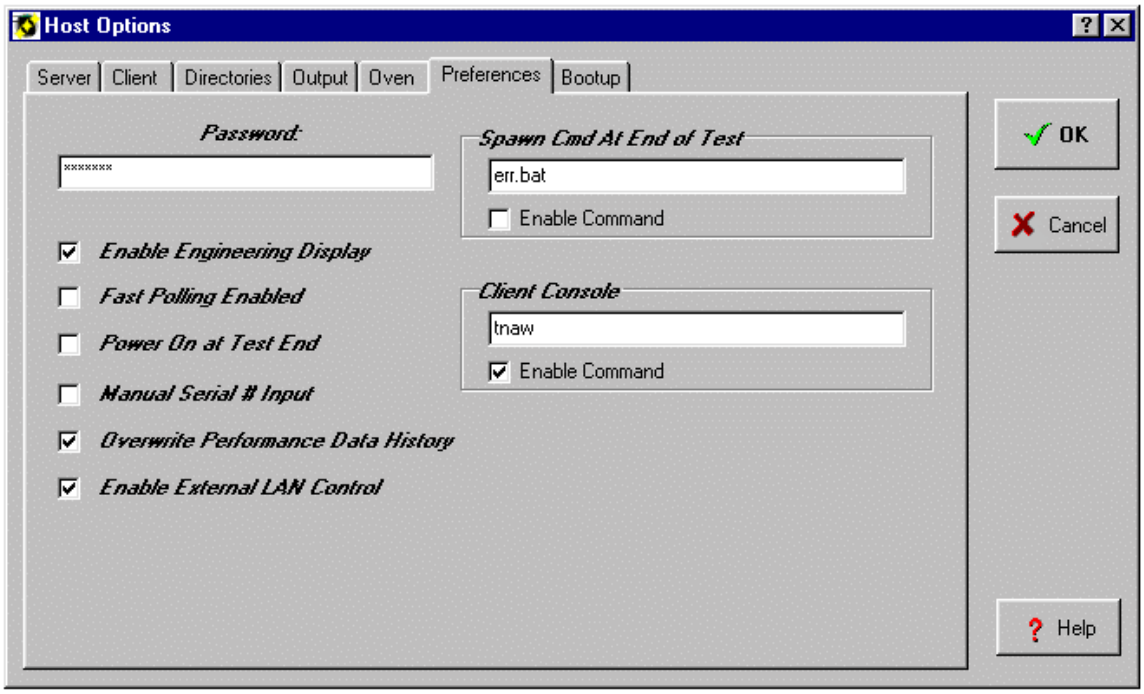


Figure 8 - Preferences

Password – Allows the user to password protect the system setup menus. The password can be any string, and is not case sensitive.

Enable Engineering Display – Shows / hides the main engineering display window.

Fast Polling Enabled – When enabled, this forces the polling priority to be given to the focused port. This is normally not required.

Power On at Test End – When this checkbox is enabled, the power to the DUT will remain ON after the test has been completed. This is useful for debugging drive problems dynamically.

Manual Serial # Input – When this checkbox is enabled, the Host software will prompt the user for a serial number when a test is started. This is used when the DUT may not contain a serial number.

Overwrite Performance History Data – Allows the system to overwrite .PRF file instead of appending (default).

Enable External LAN Control – This features allows the Host to be controlled by an external

application via the Internet. A separate document describes this type of remote control.

Spawn Command After Script Complete - This section determines external spawn command behavior.

- The spawn command string contains the path and filename and command line arguments of an executable file that may be "spawned" after a test script has been completed.

- ***Enable Command*** - This checkbox must be checked to enable this function.

Client Console – This function allows the user to launch a console window that replicates the client's text window. This method avoids having to connect a monitor to the client's video connector in order to view client video. Note also that the keyboard is redirected so this function can interact with the client.

The edit box displays the utility used for this purpose (usually TNAW.EXE). The **enable check box** will enable the function inside of the client GUI pull-down menu. See also Appendix C for more information.

5.1.7 Bootup Options

The Bootup options menu is currently not used.

5.2 OPERATION

The following section describes the steps to control the test state of each *test port* in the system.

5.2.1 Port Icon Definitions

The view pane on the left side of the GUI is painted with icons representing the current state of each test port in the system.

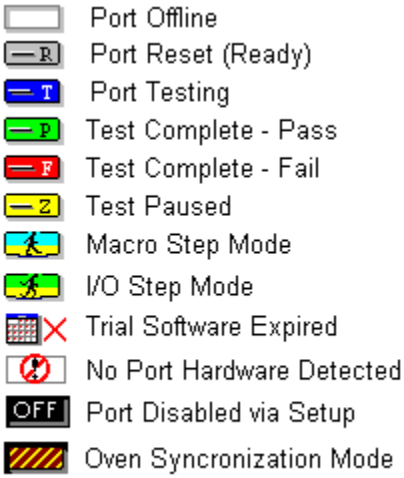


Figure 9 - Port Icon Definitions

Various operations can be performed on one or more ports. Individual ports are selected by simply clicking on the port icon in the rack view screen on the left side of the GUI.

Multiple ports are selected as follows:

1) Click on the **STARTING** port.

2) Hold down the **SHIFT** key and click on the **END** port. This will draw grouping rectangles around each port that is in the group.

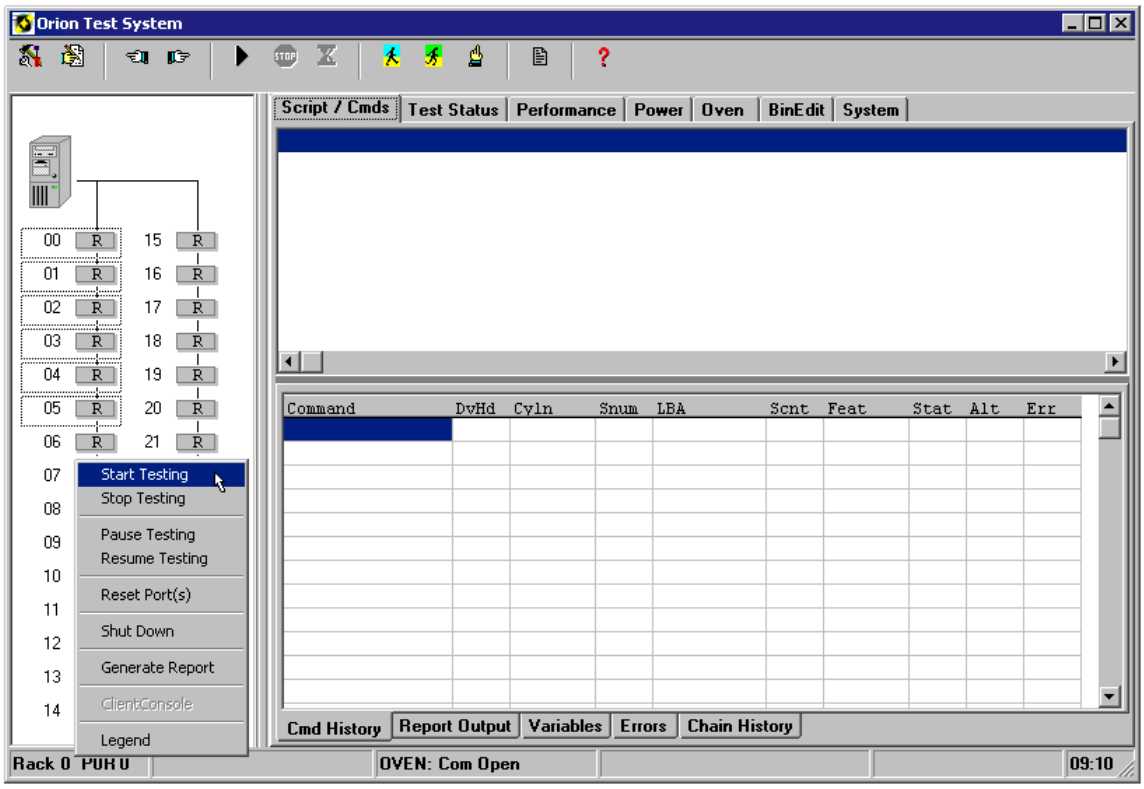


Figure 10 - Port Group Selection

5.2.2 Resetting Ports

To reset ports, first select the port(s) to be reset. Then click the **Right** mouse button over the rack view screen to bring up a popup menu. *Click on Reset Port(s)*.

The port icons associated with the selected ports should now display the **Gray** Reset icon.

5.2.3 Starting a Test

Testing is started by first selecting the port(s) to start a test on. Then click the **Right** mouse button over the rack view screen to bring up the port operation popup menu. Click on *Start Test* and a file select dialog box will appear to allow selection of the script file to be executed on the selected port(s). Select the script file and click Open to start the testing.

The port icons associated with the selected ports should now display the **BLUE** Testing icon.

The currently selected port (the one with the select rectangle) will provide information to paint the various engineering mode screens.

5.2.4 Stopping a Test

Testing is stopped by first selecting the port(s) to stop. Then click the **Right** mouse button over the rack view screen to bring up a popup menu. *Click on Stop Test*.

The port icons associated with the selected ports should now display a **RED** Fail icon to indicate that the user terminated the test.

5.2.5 Pausing and Resuming a Test

Testing is paused by first selecting the port(s) to pause. Then click the **Right** mouse button over the rack view screen to bring up a popup menu. *Click on Pause Test*.

The port icons associated with the paused ports should now display a **YELLOW** Pause icon.

The testing can now be resumed by selecting the port(s) to resume. Then click the **Right** mouse button over the rack view screen to bring up a popup menu. *Click on Resume Test*.

5.3 ORION HOST GUI

The right side of the Orion HOST GUI is a tabbed page control used to display various pages of status information. These pages represent the status of the currently selected port in the RACK VIEW pane (port with the focus rectangle around it). To select a port to view, simply click on the desired port to display its status.

5.3.1 Script / Commands Page

The Script / Cmds page has two display panes. The upper pane is a display of the currently executing script file. The current line being executed is highlighted. The lower pane is divided into several tabbed windows and is described below. A splitter between the two panes can be used to view more of one pane versus the other.

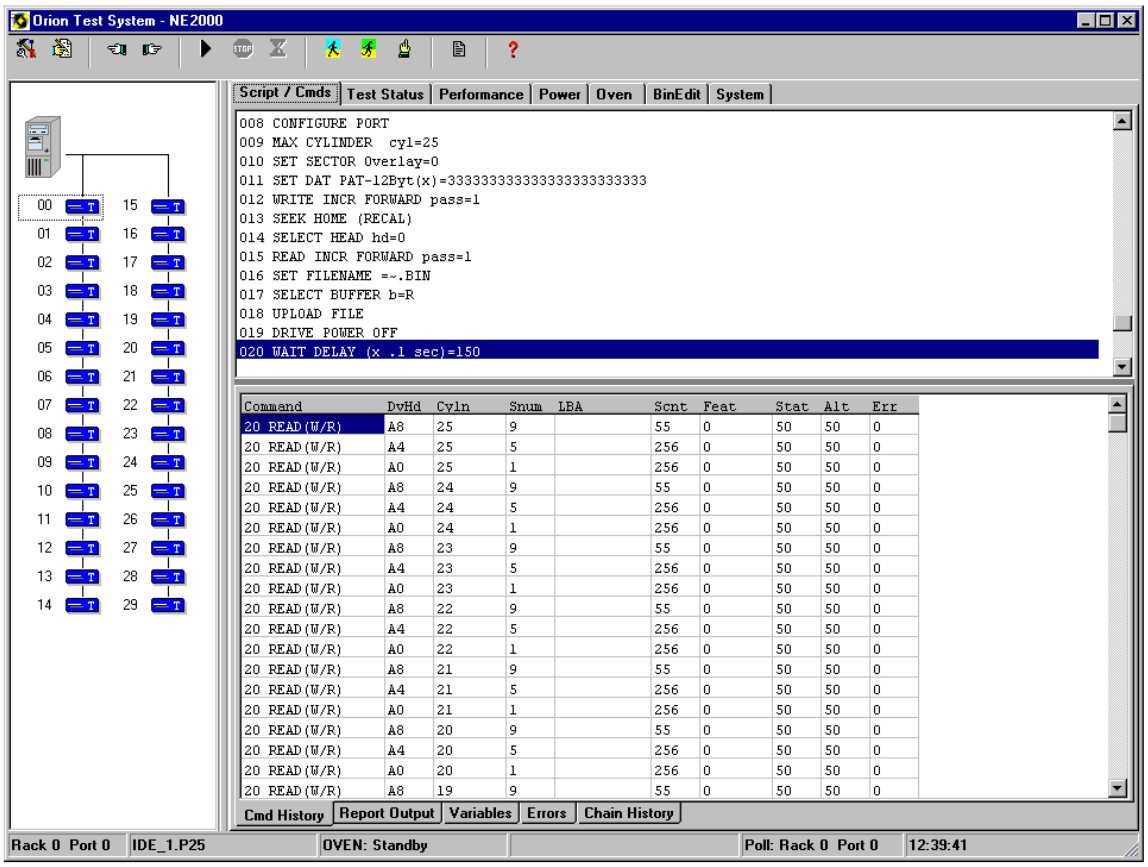

Orion Test System - NE2000
Script / Cmds
Test Status
Performance
Power
Oven
BinEdit
System
008 CONFIGURE PORT
009 MAX CYLINDER cyl=25
010 SET SECTOR Overlay=0
011 SET DAT PAT-12Byt(x)=333333333333333333333333
012 WRITE INCR FORWARD pass=1
013 SEEK HOME (RECAL)
014 SELECT HEAD hd=0
015 READ INCR FORWARD pass=1
016 SET FILENAME =~.BIN
017 SELECT BUFFER b=R
018 UPLOAD FILE
019 DRIVE POWER OFF
020 WAIT DELAY (x .1 sec)=150
Command DvHd Cyln Snum LBA Scnt Feat Stat Alt Err
20 READ(W/R) A8 25 9 55 0 50 50 0
20 READ(W/R) A4 25 5 256 0 50 50 0
20 READ(W/R) A0 25 1 256 0 50 50 0
20 READ(W/R) A8 24 9 55 0 50 50 0
20 READ(W/R) A4 24 5 256 0 50 50 0
20 READ(W/R) A0 24 1 256 0 50 50 0
20 READ(W/R) A8 23 9 55 0 50 50 0
20 READ(W/R) A4 23 5 256 0 50 50 0
20 READ(W/R) A0 23 1 256 0 50 50 0
20 READ(W/R) A8 22 9 55 0 50 50 0
20 READ(W/R) A4 22 5 256 0 50 50 0
20 READ(W/R) A0 22 1 256 0 50 50 0
20 READ(W/R) A8 21 9 55 0 50 50 0
20 READ(W/R) A4 21 5 256 0 50 50 0
20 READ(W/R) A0 21 1 256 0 50 50 0
20 READ(W/R) A8 20 9 55 0 50 50 0
20 READ(W/R) A4 20 5 256 0 50 50 0
20 READ(W/R) A0 20 1 256 0 50 50 0
20 READ(W/R) A8 19 9 55 0 50 50 0
Cmd History
Report Output
Variables
Errors
Chain History
Rack 0 Port 0
IDE_1.P25
OVEN: Standby
Poll: Rack 0 Port 0
12:39:41

Figure 11 - Script / Commands Page

5.3.1.1 Command History Page

The Command History page (see above, section 5.3.1 – lower pane) is used to display a list of recent low level commands that were sent to the device under test. The format of this page is interface dependent. The header at the top of the page will change depending on the interface type being tested. The header shown in 5.3.1 is an IDE interface type.

5.3.1.2 Report Output Page

The Report Output pane is a text pane that will show the results of the last report generated by the system for the selected port. The report is the same file that would output to the system printer if enabled.

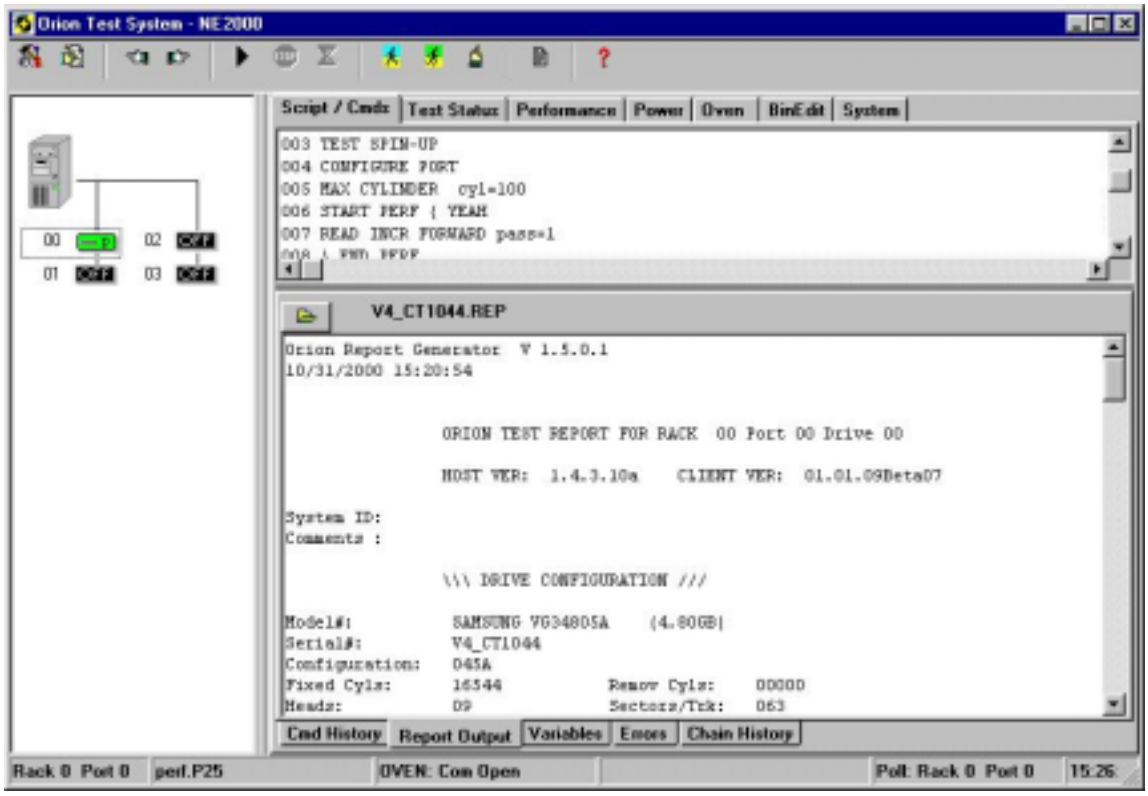


Figure 12 - Report Output Page

5.3.1.3 Variables Page

The Variables pane displays the variables that are used by the running script and their values. The values are updated dynamically whenever they change.

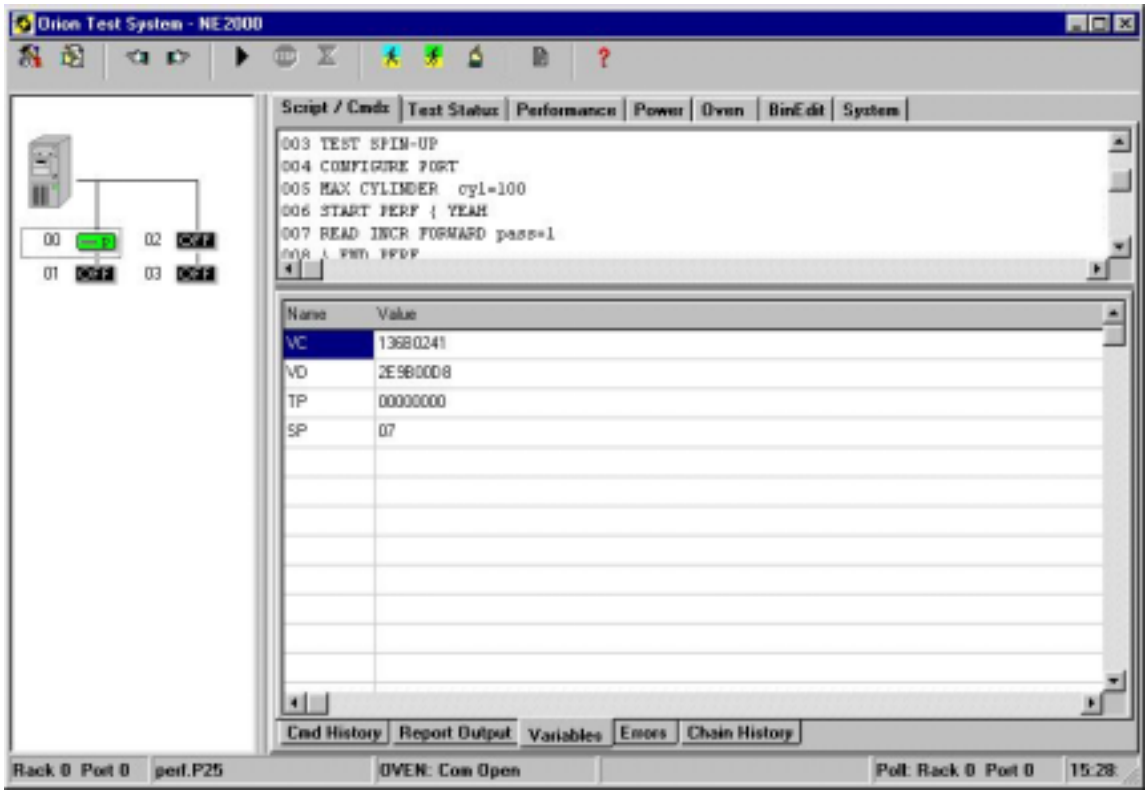


Figure 13 - Variables Page

5.3.1.4 Errors Page

The Error pane displays any errors that may have been collected by the client test port. The errors are displayed in the same format as they appear on the report file.

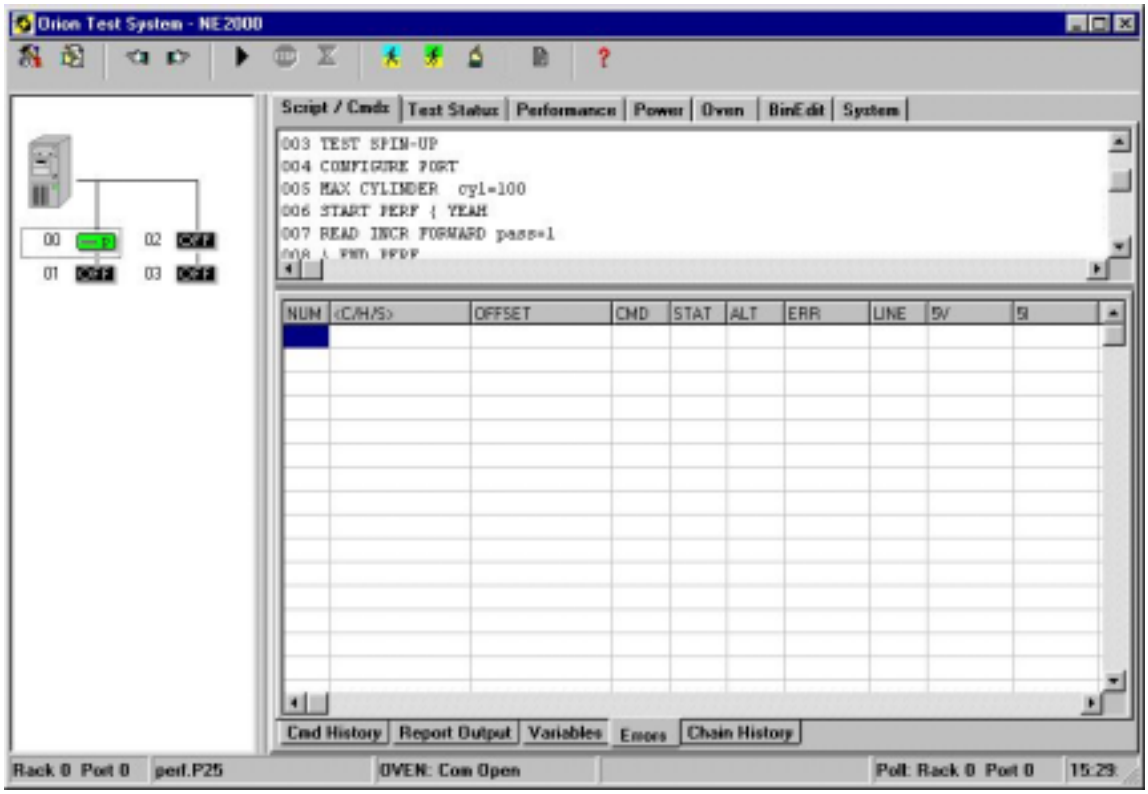


Figure 14 - Errors Page

5.3.1.5 Chain History Page

The Chain History pane displays the execution history of the root script and any scripts that may have been called from the root or any other script. The display also shows the time when the chain began and when it ended. Note that the size of the chain history is limited to 256 entries. Also note that the chain FIFO size is reduced to 50 entries on the report printout.

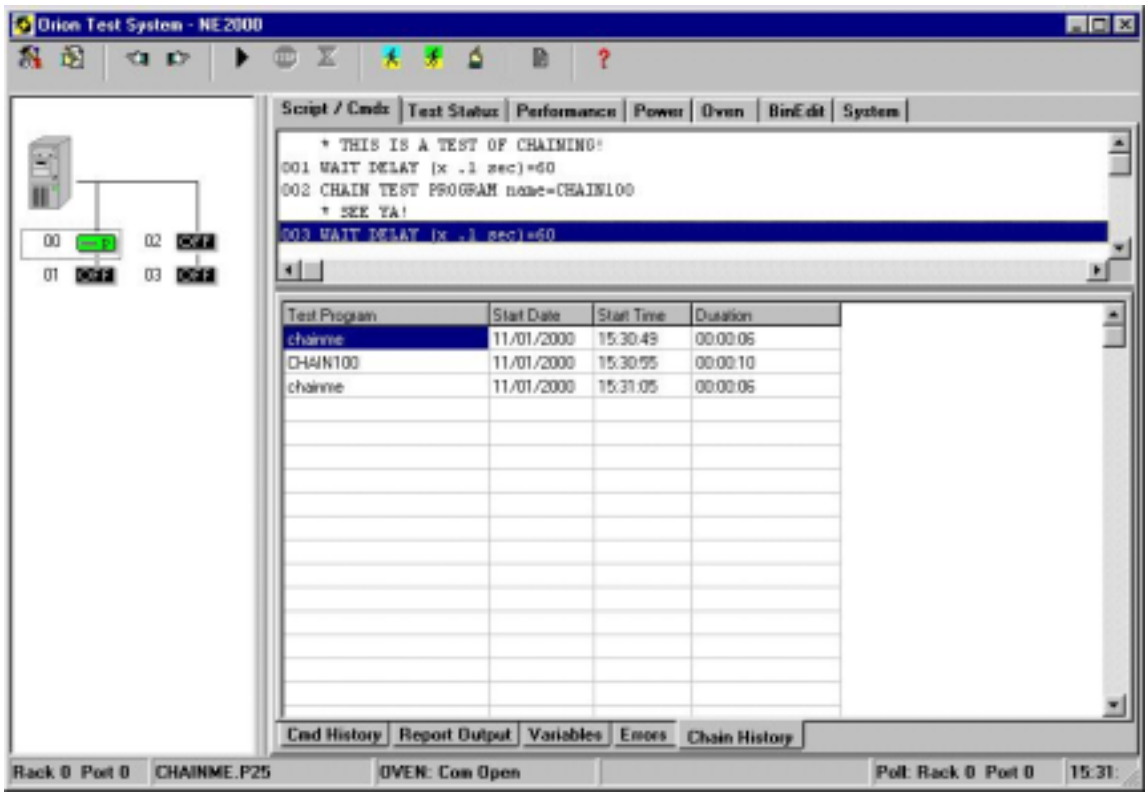


Figure 15 - Chain History Page

5.3.2 Test Status Page

The Status Page contains the following:

1) Various strings such as serial number, model number, etc.

2) Device geometry.

3) Error counters.

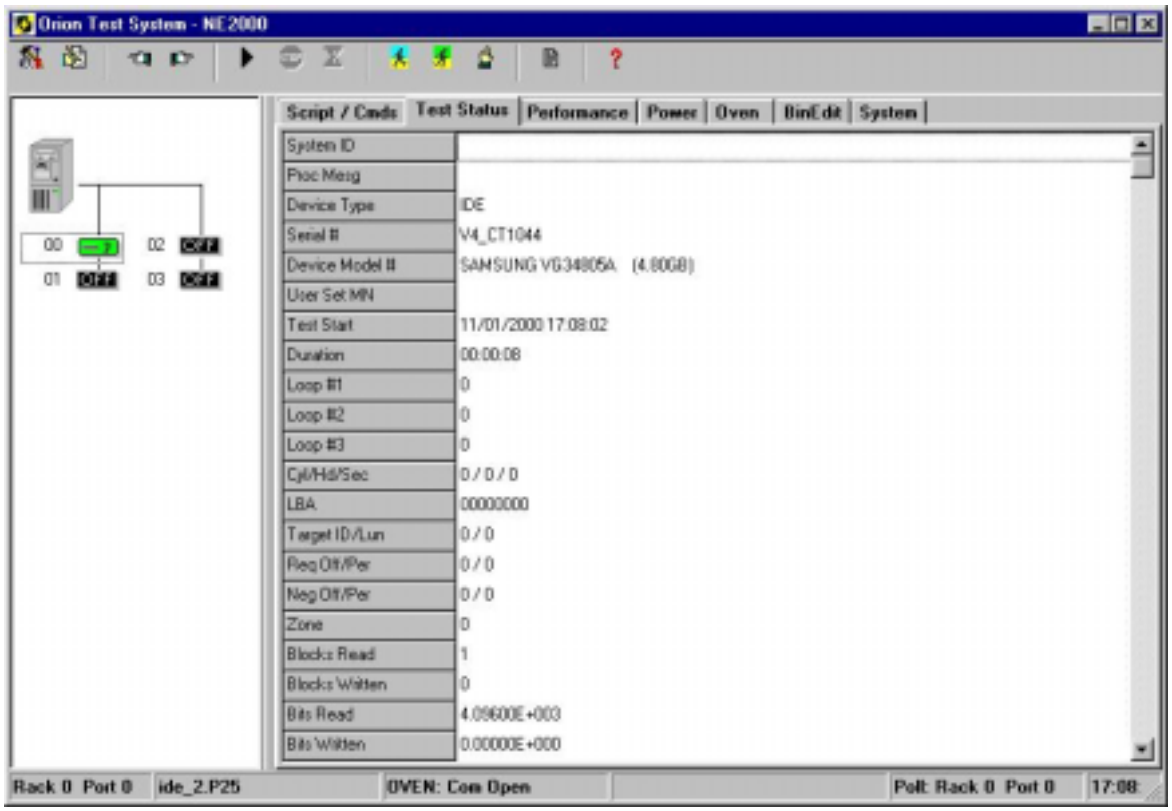


Figure 16 - Test Status Page

5.3.3 Performance Page

This page contains specific information about seek timing and data transfer test results. A graph on this page displays results of the last performance test for the selected port.

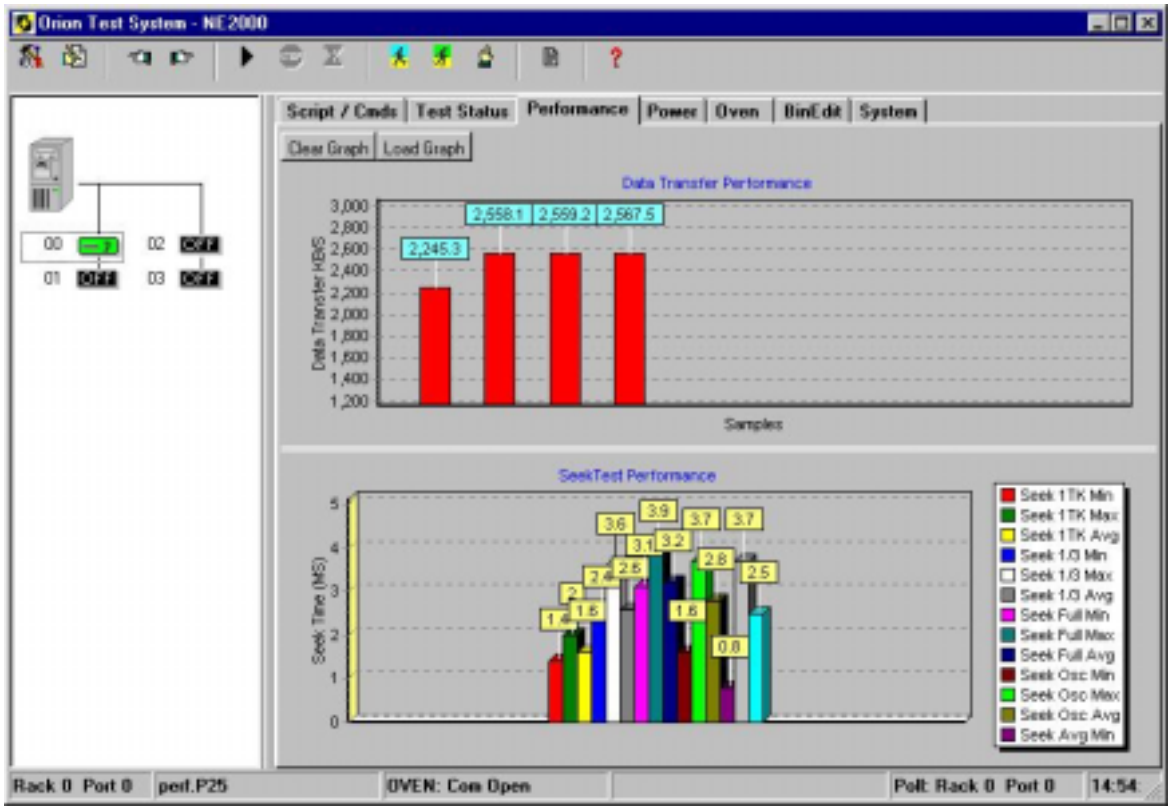


Figure 17 - Performance Page

Data Performance Chart

The Data Performance Chart shown above is an example of a test that measured data throughput four different times for the same test sequence. Each occurrence of this test produces a new bar on this chart, which displays the data throughput. Thus many samples can be made and compared manually. Also, the samples do not have to be the same test sequence – i.e. each bar could represent a different type of test: random vs. sequential, read vs. write, etc.

Note that each bar on this chart is displayed in units of Kilobytes per second. The results are dependent on how the actual performance test is written (see section 6.2.1.13).

Seek Test Performance Chart

The Seek Test Performance Chart produces a bar for each element of the each of the various seek

timing tests. The bar represents the measured time for that particular measurement in milliseconds.

Note that all charts and graphs in the Orion test system can be zoomed and scrolled. To zoom in on a particular area, click the ***left*** mouse button and drag the mouse from ***left to right*** to draw the rectangle to be viewed. To scroll, click the ***right*** mouse button and drag the graph in the desired direction. To zoom out to normal, click the ***left*** mouse button and draw any rectangle from ***right to left*** (reverse of the zoom-in operation).

5.3.4 Power Display Page

This page displays information about the current device voltage and current as well as a graph of the last power profile 'snapshot' that was done.

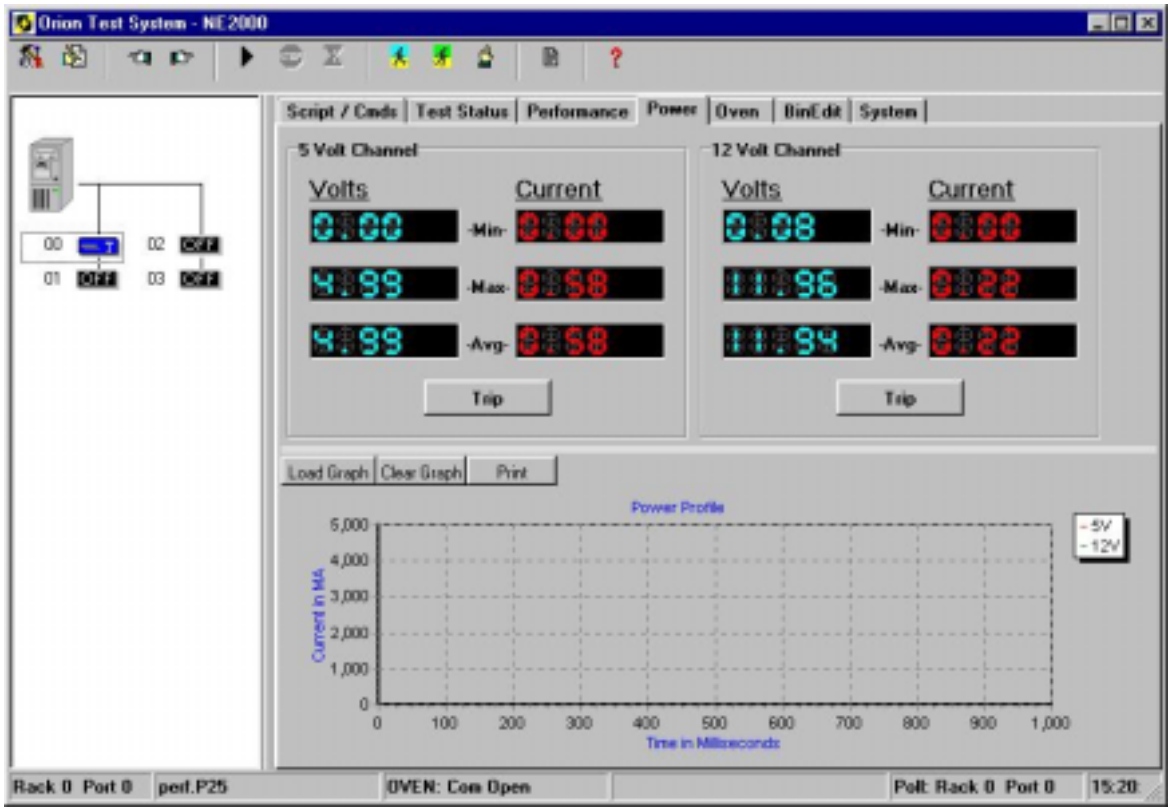


Figure 18 - Power Display Page

The 5 Volt and 12 Volt channel displays show Min, Max, and Averaged values for the port that is currently selected (focused). These values may be reset to begin sampling new Min, Max, and Average values using the ***Trip*** button located under each channels display meters.

The values displayed in the channel display meters are sampled by the client every three seconds on average. This time may vary under some conditions such as TEST DRIVE SPINUP where the client is

busy with low level functions.

The Power Profile graph displays a snapshot of the last PROFILE CURRENT command issued in the test script. Each time this command is executed, two colored line graph data sets are produced – one for each channel. The graph may be saved or printed. Graphs that were archived can be re-loaded for later analysis.

5.3.5 Oven Status Page

This page displays a line graph showing a history of the temperature and humidity if a oven controller is connected to the system and oven control is enabled in the setup menu. Also displayed are the current oven script and the time that the script has been executing. Note that temperature is always reported in degrees Celcius.

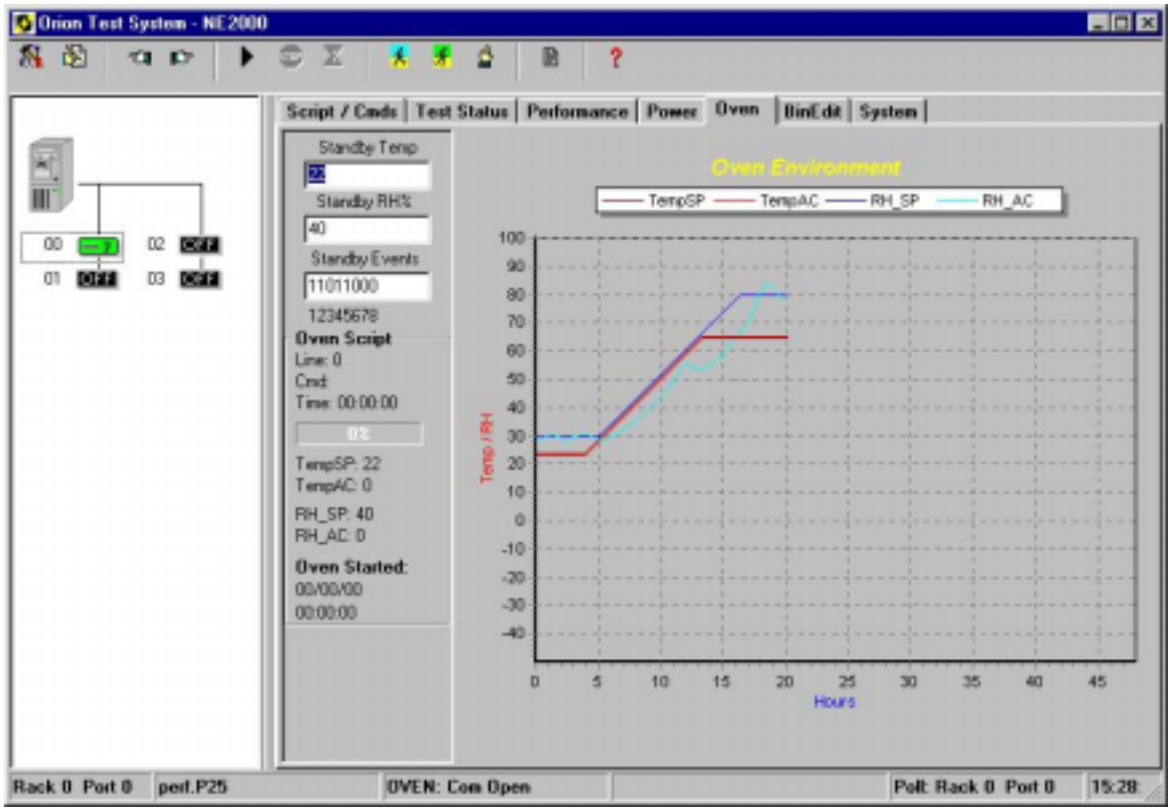


Figure 19 - Oven Status Page

Note that all charts and graphs in the Orion test system can be zoomed and scrolled. To zoom in on a particular area, click the ***left*** mouse button and drag the mouse from ***left to right*** to draw the rectangle to be viewed. To scroll, click the ***right*** mouse button and drag the graph in the desired direction. To

zoom out to normal, click the ***left*** mouse button and draw any rectangle from ***right to left*** (reverse of the zoom-in operation).

5.3.6 Binary Editor Page

See section 5.4.3 for operational details.

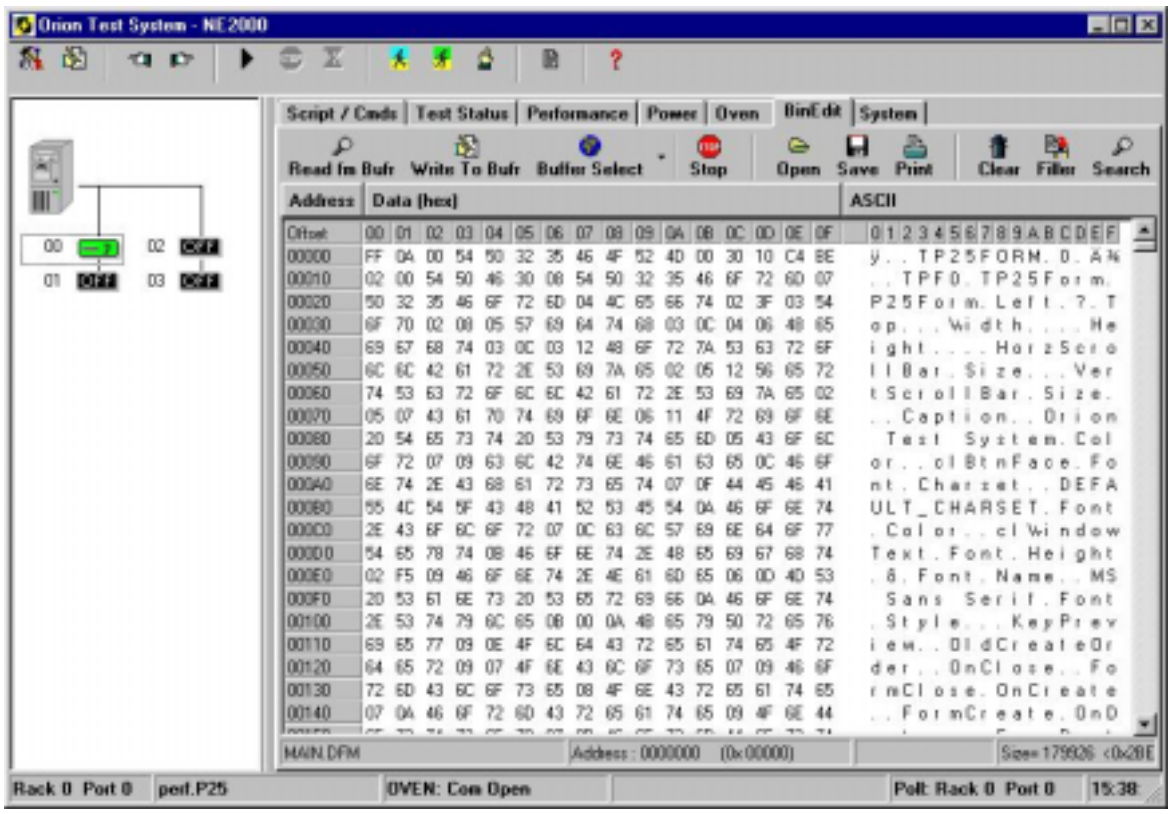


Figure 20 - Binary Editor Page

5.3.7 System Page

The system page is used for debug purposes and displays system resources.

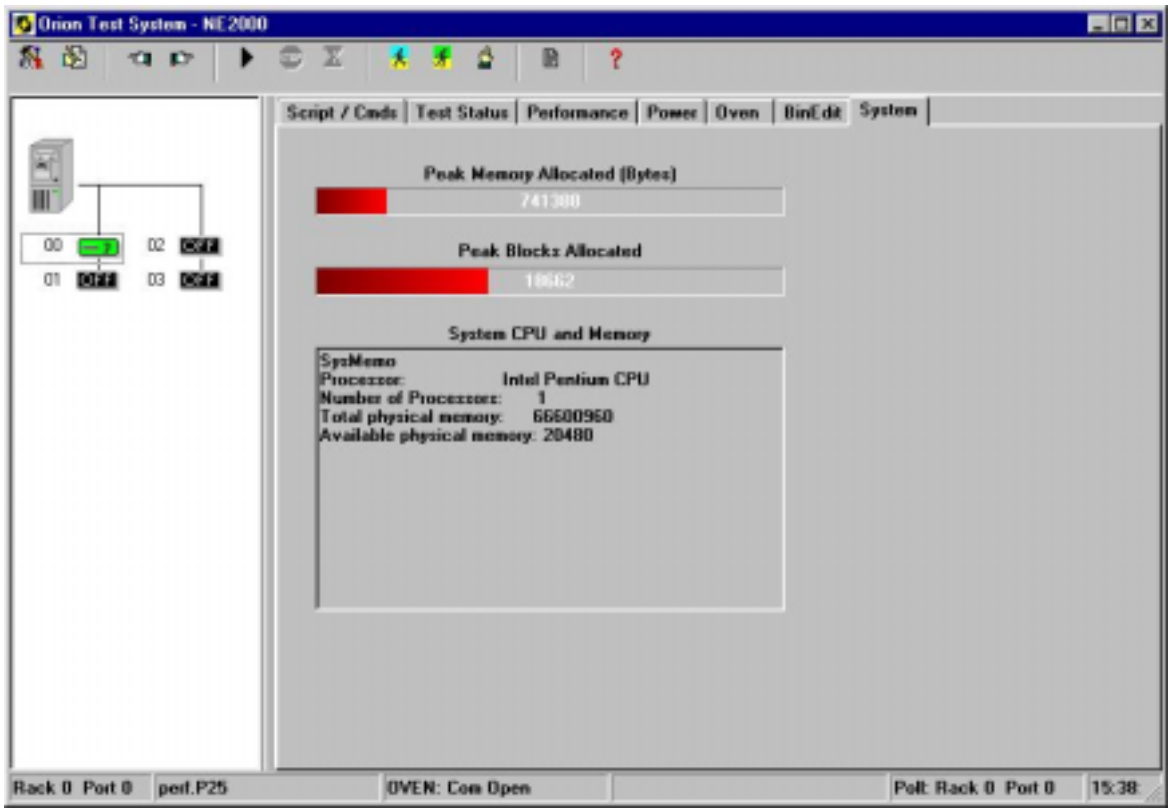


Figure 21 - Systems Page

5.3.8 Toolbar Buttons

Figure 22 shows the various toolbar buttons on the main menu:

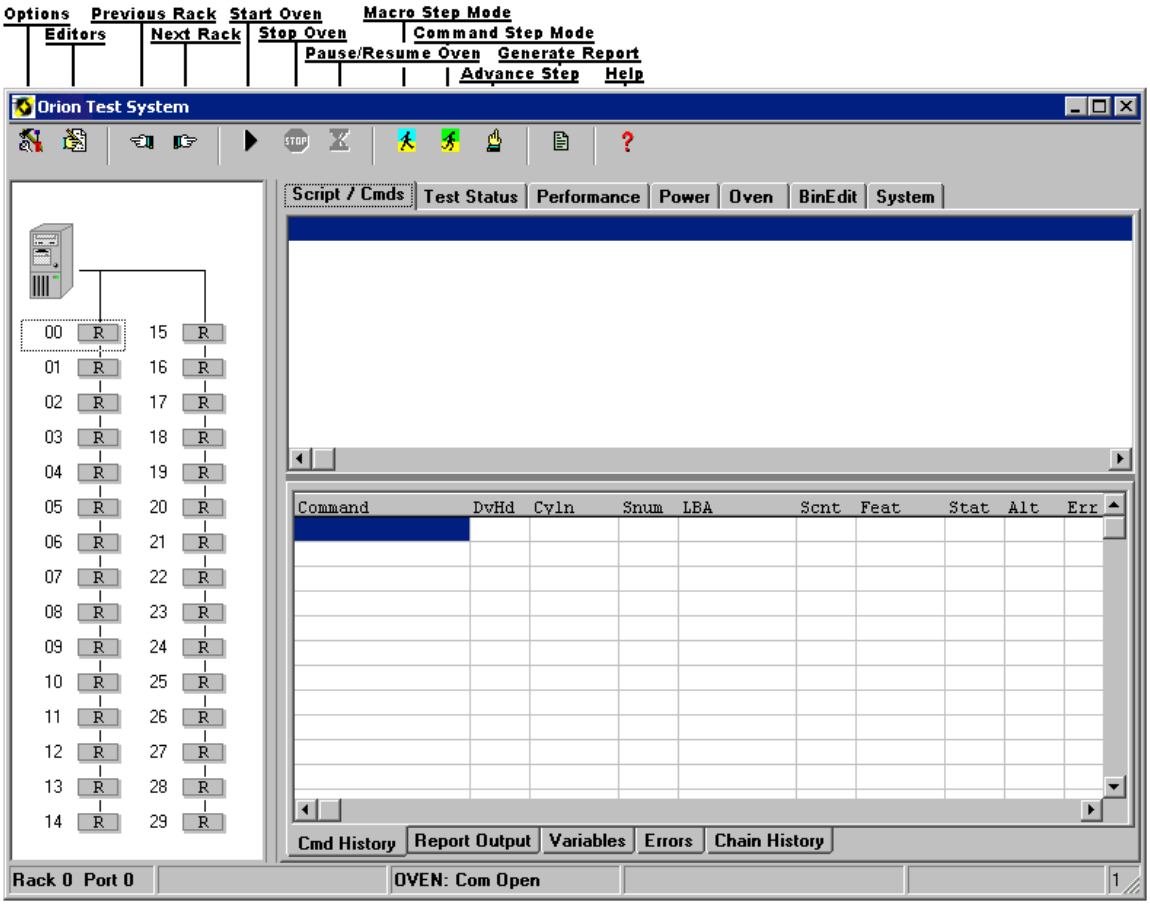
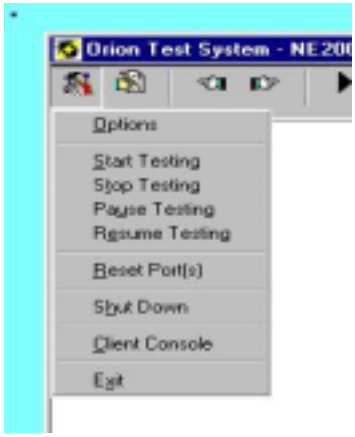


Figure 22 - Toolbar Buttons

5.3.8.1 Options Button

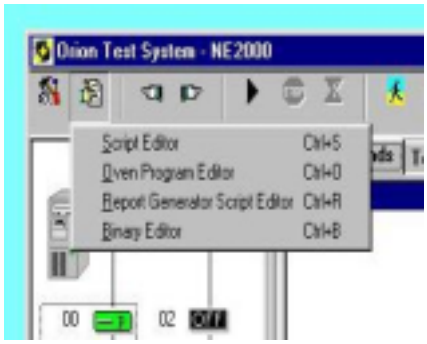
The OPTIONS button will create a dropdown menu as follows:



These menu items are discussed in section 5.1.

5.3.8.2 Editors Button

The EDITORS button will produce a dropdown menu as follows:



These menu items are discussed in the Utilities section 5.4.

5.3.8.3 Previous Rack Button

The PREVIOUS RACK button is used to toggle to the previous rack view. Only one rack is visible at any one time on the rack view screen. This button will only function if the selected rack is > 0. The maximum number of racks per server is 4.

5.3.8.4 Next Rack Button

The NEXT RACK button is used to toggle to the next rack view. Only one rack is visible at any one time on the rack view screen.

5.3.8.5 Start Oven Button

The START OVEN button is used to manually start an oven script running. See also Oven Description in section 8.0

When this button is clicked, a **Oven File dialog box** is shown which allows the user to select an oven script file from the Oven Script directory. If a script file is selected, the Orion software will begin to execute this script. See also section 8.2 (oven script editor).

5.3.8.6 Stop Oven Button

The STOP OVEN button is used to manually halt any running oven script. This button does nothing if no script is currently executing.

5.3.8.7 Oven Pause/Resume Button

The OVEN PAUSE/RESUME button is used to pause or resume oven script execution. If a script is running, it may be put into a pause mode by clicking this button. The button will be displayed in a DOWN state indicating that the oven script has been paused.

Clicking this button again will restart a paused script. The button will now be displayed in the UP state indicating that the oven script is executing.

5.3.8.8 Macro Step Mode Button

A running test script may be placed in a single step mode called MACRO STEP mode. The macro step mode allows the user to execute one script instruction at a time by clicking the ADVANCE STEP button.

The MACRO STEP mode is entered by clicking this button once. The button will then be displayed in the DOWN state indicating that the MACRO STEP mode is active. Clicking this button again will place the test script back in its normal RUN mode. See also variable MS.

5.3.8.9 Command Step Mode Button

A running test script may be placed in a single step mode called COMMAND STEP mode. The command step mode allows the user to execute one low level device command at a time by clicking the

ADVANCE STEP button. A single script command may contain many low-level device commands.

The COMMAND STEP mode is entered by clicking this button once. The button will then be displayed in the DOWN state indicating that the COMMAND STEP mode is active. Clicking this button again will place the test script back in its normal RUN mode. See also variable MS.

5.3.8.10 Advance Step Button

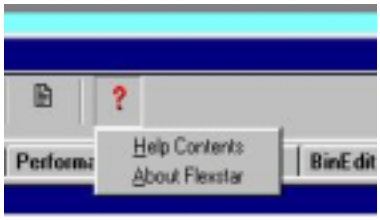
The ADVANCE STEP button is used to cause the test script to execute the next instruction. This button is only used when either the MACRO or COMMAND step mode is active.

5.3.8.11 Generate Report Button

The GENERATE REPORT button is used to force an on-the-fly image update and report generation. If the report output to system printer option is enabled in the setup menu, the report will also be output to the selected system printer.

5.3.8.12 Help Button

The HELP button creates the following dropdown menu:



Selecting HELP opens the associated Windows help file for the Orion Host. The help file will be initially be opened to the contents page.

Selecting ABOUT causes an about dialog box to appear which indicates some system information and the current version number of the Orion Host application.

5.4 UTILITIES

5.4.1 Test Script Editor

The Orion Script Editor is a script editing tool that is used to create or modify Flexstar 'P25' test scripts. The script editor is normally launched from within the Orion Test System environment but can also be operated as a stand-alone program. The filename of this program is O25EDIT.EXE and is usually located in the same directory as the O25HOST.EXE program.

See section 6.1 *USING THE SCRIPT EDITOR*.

5.4.2 Oven Script Editor

See section 8.2 *OVEN SCRIPT EDITOR*.

5.4.3 Binary Editor

The Orion test system contains a built-in binary editor, which allows access to the client buffers as well as normal system files. Operation is detailed below.

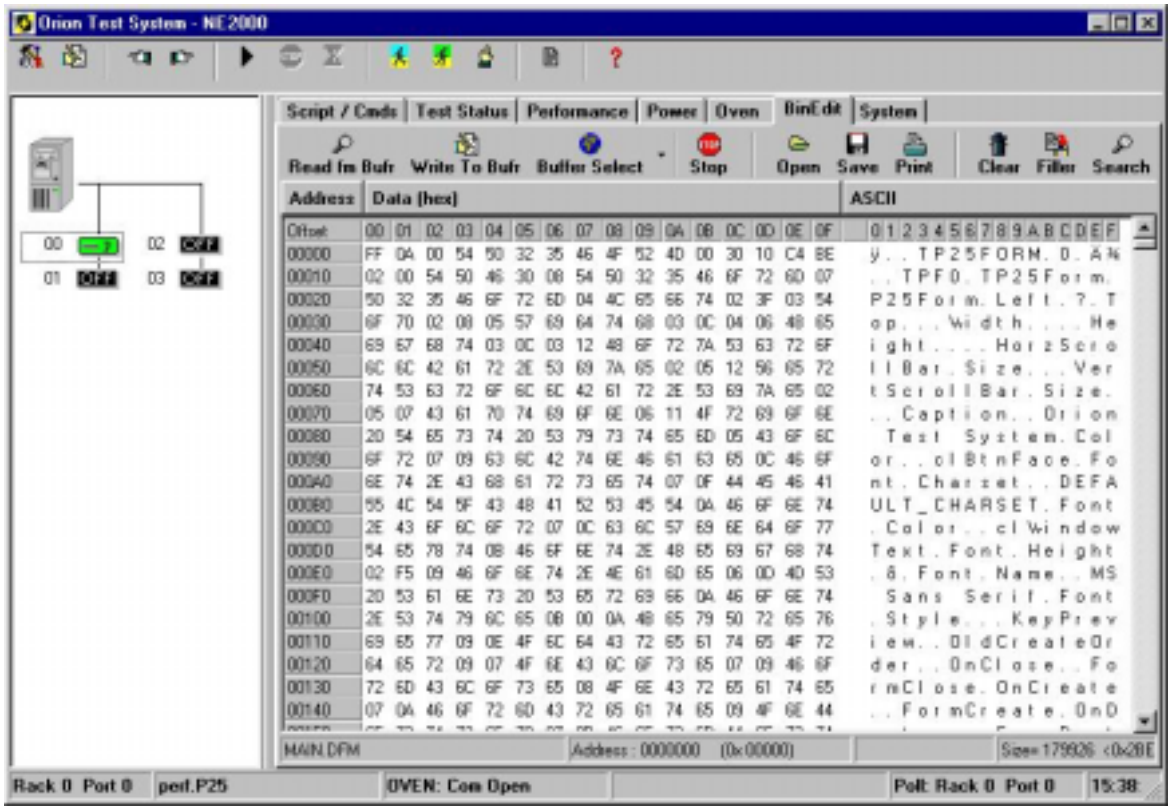


Figure 23 - Binary Editor Description

5.4.3.1 Opening a File

To open a file for display and editing, click on the *Open* button. This will invoke a file open dialog box that allows the user to browse for the desired file. Upon selecting a file, the file contents are displayed in hexadecimal and ASCII formats. The hexadecimal portion of the screen may be edited. The file name and size are displayed on the bottom of the window.

5.4.3.2 Saving Editor Contents to a File

To save the editor contents to a file, click the *Save* button. This will invoke a save file dialog box which allows you to save the contents to a named file.

5.4.3.3 Loading from Client Buffer

To load the editor contents from a specified client buffer, first click the Buffer Select button and choose which buffer to load from. Then click the *Read fm Bufr* button to perform the upload from the client. Depending on the buffer selected, the size will either be 512K bytes or 64K bytes.

5.4.3.4 Saving to a Client Buffer

To store the contents of the editor into the client's buffer, first click the Buffer Select button and choose which buffer to store to. Then click the *Write To Bufr* button to perform the download to the client.

5.4.3.5 Printing the Editor Contents

The contents of the editor may be sent to the system printer. This is done by clicking the *Print* button. This will invoke a print preview screen displaying how the data will look when it is printed out. If printing is to continue, you can select print from the preview screen. This also allows the user to perform a partial print and generate multiple copies, etc.

5.4.3.6 Buffer Selection

There are several client buffers that can be selected via the ***Buffer Select*** button. They are as follows:

- Buffer 0 to 3 – Four 64K general-purpose buffers for saving disk data, etc.
- Read Buffer – This 512K buffer contains the contents of the last READ operation (data transferred from the DUT).
- Write Buffer – This 512K buffer contains the data to be written to the DUT.

5.4.3.7 Other Features

Clear – The clear button will clear the contents of the editor screen.
Filler – The filler button allows the user to fill the editor with user defined patterns.
Search – Allows the user to search for hexadecimal patterns.
Stop – The user can abort a long operation by clicking the stop key.

5.5 DATABASE OUTPUT

Database output files are generated by the software at the time that the testing has completed or real time as a script command. Output files are sent to the Output File destination path set in the Directories page of the setup menu. The database files are mixed ASCII and binary data. The built-in report generator tool is designed to parse this data and provide custom ASCII report files. See also Appendix A.

5.5.1 Legacy Files

The following files are generated to maintain legacy (FS2500) compatibility:

Image files (*.IMG) - Binary database output files are created with the P25 style naming convention RxxPxxDx.IMG where Rxx is a **rack** number, Pxx is the **port** number in hex, and Dx is the **drive** number. The map of this database file is available in the file called IMG800.FYL. See also Appendix A.

Image files are located in the directory **\Program Files\Flexstar\Orion** by default but the path can be changed through the setup menu.

Data files (*.DAT) – Same content as image files except that the files are named using the first 8 characters of the device's serial number.

Text report files (*.REP) - Text (ASCII) report files are generated using a script driven report generator which can create legacy style reports or can be customized for specific needs.

SCSI sense files (*.SNS) - These files contain a dump of SCSI sense data when a fatal error occurs on a SCSI device.

6.0 Orion Script Language

6.1 USING THE SCRIPT EDITOR

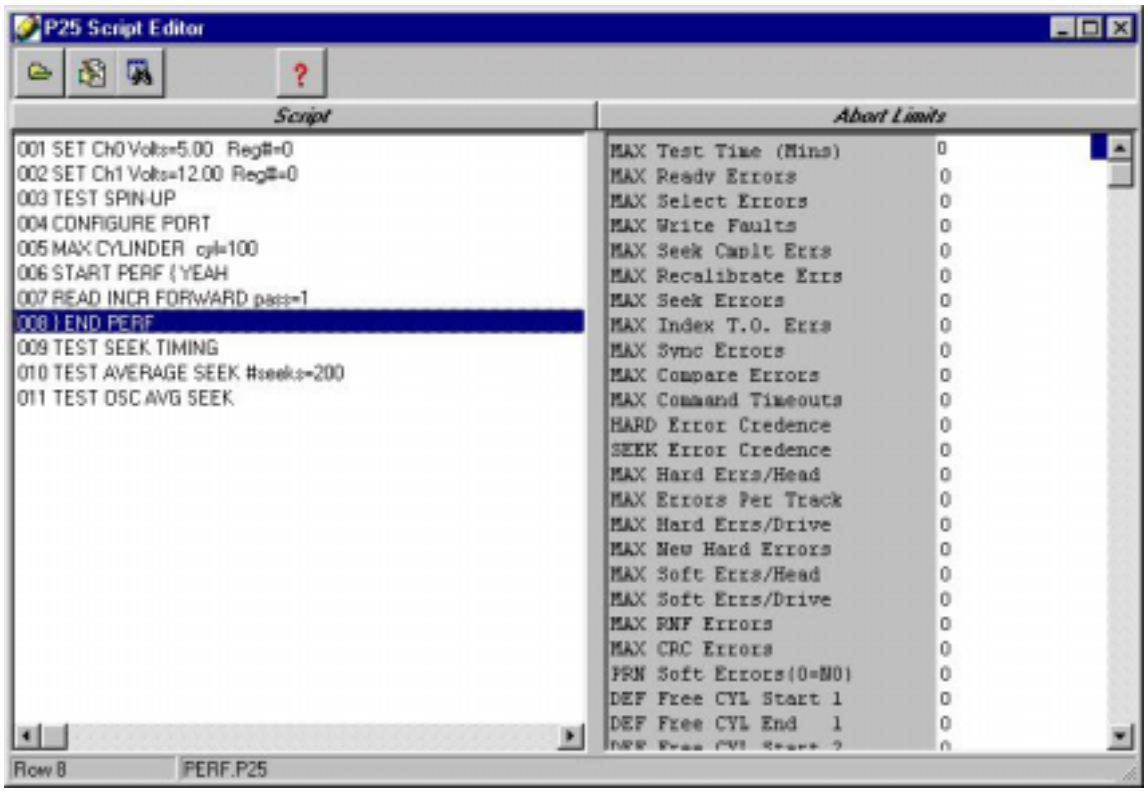

P25 Script Editor
Script
Abort Limits
001 SET Ch0 Volts=5.00 Reg#=0
002 SET Ch1 Volts=12.00 Reg#=0
003 TEST SPIN-UP
004 CONFIGURE PORT
005 MAX CYLINDER cyl=100
006 START PERF (YEAH
007 READ INCR FORWARD pass=1
008) END PERF
009 TEST SEEK TIMING
010 TEST AVERAGE SEEK #seeks=200
011 TEST OSC AVG SEEK
MAX Test Time (Mins) 0
MAX Ready Errors 0
MAX Select Errors 0
MAX Write Faults 0
MAX Seek Cmplt Errs 0
MAX Recalibrate Errs 0
MAX Seek Errors 0
MAX Index T.O. Errs 0
MAX Sync Errors 0
MAX Compare Errors 0
MAX Command Timeouts 0
HARD Error Credence 0
SEEK Error Credence 0
MAX Hard Errs/Head 0
MAX Errors Per Track 0
MAX Hard Errs/Drive 0
MAX New Hard Errors 0
MAX Soft Errs/Head 0
MAX Soft Errs/Drive 0
MAX RNF Errors 0
MAX CRC Errors 0
PRN Soft Errors(0=NO) 0
DEF Free CYL Start 1 0
DEF Free CYL End 1 0
Row 8
PERF.P25

Figure 24 - Script Editor

6.1.1 Starting the Editor

The script editor is started by clicking the EDITORS menu button on the main toolbar. Select "Script Editor" to launch the application. The script editor will be shown using the test script directory specified in the system setup menu DIRECTORIES (5.1.3).

6.1.2 Creating a New Script

To create a new script, click the ***Files*** button and select ***New*** on the dropdown menu. This will clear the script editing area and allow new script commands to be entered.

Script commands can be added in two ways. The **first** method is to click the ***Edit*** button and choose ***Insert Cmd*** from the dropdown menu. The **second** method is to right click the mouse button inside of the script editing area to obtain the dropdown menu.

Script Size limitations

Binary script files are limited to a maximum size of 4096 bytes. This limitation was inherited from legacy equipment and maintained in order to provide full compatibility with existing legacy scripts. Since command size varies from one byte to 34 bytes, the maximum number of lines possible is 4096 but is more likely to be limited to around 500 lines (averaging 4 bytes per command).

Scripts can also be saved without comments which are much more efficient than saving with comments (see 6.1.7).

Note that with the script editor 2.0, the size of the script has been increased to 65K.

6.1.3 Inserting New Script Commands

When ***Insert Cmd*** is selected from the ***Edit*** dropdown menu, a dropdown menu is displayed which is used to select commands by group. Scroll to the desired command group, which will then display a second dropdown menu of the commands available within that group. Click on the desired command to add it to the script being created. If the new command contains parameters, the editor will highlight the new command and allow editing of the parameter field.

Note that all inserted commands are inserted ***above*** the current highlighted command.

6.1.4 Modifying Parameters

Parameter fields within the script command are edited by positioning the cursor to the desired command. This is done by clicking anywhere on the script command which places the flashing cursor at the beginning of the script command.

To enter the ***Edit Parameter*** mode, press the ***Enter*** key on the keyboard or ***double click*** on the command, which then highlights the command and launches a parameter entry dialog box that contains two entry fields. If the command contains more than one parameter, the next parameter field is entered when the ***Enter*** key is pressed. When all parameters have been edited, the parameter edit dialog will be canceled and the cursor will return to the beginning of the script line.

6.1.5 Deleting Script Commands

Script commands can be deleted by positioning the flashing cursor on the command to delete and obtaining the edit dropdown menu by clicking the **Edit** button or by right clicking on the script edit area. Select **Delete Cmd** to remove the command from the script program. The DEL key can also be used to delete the current command. With script editor 2.0, blocks of script can be deleted by highlighting multiple lines and pressing the DEL key.

6.1.6 The Replace Command

The script editor also supports a REPLACE command, which allows the user to replace the currently highlighted command with another. This is analogous to first deleting a command and then inserting a

new command in that line.

6.1.7 Save Comments in Script

The SAVE COMMENTS option allows scripts to be saved without the overhead of embedded comments. If this option is checked, saved scripts will not contain any comments. This mode is the default.

6.1.8 Saving Script Programs

Script programs are saved by clicking the **Files** button on the tool bar and choosing either 'Save' (to save a previously named script) or 'Save As' (to save with a new name). Script files are saved as *.P25 files.

6.1.9 Printing Script Programs

To print the script program, click the **Files** button and select **Print.** This will launch a print preview window, which will show the script report, as it will appear on the actual printout. The user may then print the script report by clicking the Print tool bar button. Other options, such as printer setup, can also be performed via various tool bar buttons.

Note that using the save and open buttons in the print preview menu is not recommended since the preview menu stores and retrieves files using a non-standard file format.

6.1.10 Editing Abort Limits

Each script program has it's own set of unique *ABORT LIMITS* tied to the program. The Orion Script Editor displays the abort limits for each script at the right hand side of the editor. To change an abort limit value, scroll to the specific abort limit cell and then click on that cell. Then simply type in a new value for that limit.

6.1.11 Copy, Cut, and Paste

Script editor version 2.0 and above support cut and paste operations. One or more command lines can be copied, cut, or pasted into the current script.

The currently selected line(s) can be copied using the hot key combination CTRL-C. Lines can also be cut using CTRL-X. These saved lines can be pasted into the program above the currently highlighted line by pressing CTRL-V. These functions are also available in the edit dropdown menu.

6.1.12 Enhanced Script Feature

Script editor version 2.0 and above supports a large script file feature that expands a script file size from 4K bytes to 64K bytes. Since large script files are in a different format, the file extension for large script files has been changed to *.P2E. The editor will now support both small (P25) and large (P2E) file types.

6.2 SCRIPT LANGUAGE DESCRIPTION

When the user chooses the Insert Cmd or Replace Cmd function, a drop down menu of command groups will appear.

Each GROUP contains a number of actual script commands. As the cursor is moved over each group entry in the drop down menu, a command drop down sub-menu is displayed next to the GROUP menu showing each supported command. Select a command from this sub-menu.

6.2.1 COMMANDS BY GROUP

ATAPI Commands
Loop & Utility Commands
Oven Control Commands
IDE Commands
Program Control Commands
Read Commands
Write Commands
SCSI Commands
CCS Command Subgroup
Command Terminators
Set Parameter Commands
Seek Commands
Test Macro Commands
Voltage Margin Commands
Variables

6.2.1.1 ATAPI Commands

Build ATAPI Packet Command
Configure Port
Data Streaming Test
Generic Access Test
Open/Close Tray

DASP/PDIAG Timing

Test CD Transfer Rate Suite A

Test CD Access Time Suite A

1. BUILD ATAPI PACKET Cmd=[xx,xx,…,xx,T]

PARAMETER:

1) 24 digit hexadecimal (comma delimited) string plus a one character terminator (**T**).

PURPOSE: This command allows the construction of a ATAPI Packet Command of up to 12 bytes in length plus an optional COMMAND TERMINATOR (**T**) character. The ATAPI packet command can be built using variables in the same manner as the BUILD SCSI CDB command.

2. CONFIGURE PORT {ATAPI}

PARAMETER:

none

PURPOSE:

Issue this command to set the ATAPI test port parameters such as MAX LBA, device type, TOC, etc. The following sequence is used by the CONFIGURE PORT command:

TEST UNIT READY is issued twice to clear any power up CHECK CONDITIONS. If the device reports a CHECK CONDITION, a REQUEST SENSE is issued to clear the error condition.

ATAPI IDENTIFY command is issued to get values, S/N, buffer type, buffer size, firmware revision, model number, capabilities, and timing information.

INQUIRY command is issued to retrieve data from the device.

READ CAPACITY command is issued to read the devices Logical Block Address (LBA). This command is done twice - first to obtain the devices maximum LBA which is placed in the MX variable. The second time to obtain the highest LBA on cylinder zero (blocks per cylinder).

MODE SENSE is issued with a 0x3F for the page code to retrieve data from the device.

READ TABLE OF CONTENTS is issued in both Logical Block Addressing (LBA) and Minute/Second/Frame (MSF) mode to display Table of Contents (TOC) information in the report file. A CD must be inserted in the device for this to work properly.

RECAL command is issued to the device.

SEE ALSO:

CONFIGURE PORT (SCSI)

CONFIGURE PORT (IDE)

3. DATA STREAMING TEST=[xxxxxxxxxxxx]

PARAMETER:

1) 12 character ASCII string.

PURPOSE:

Legacy system only – Orion does not support this command.

4. GENERIC ACCESS TEST=[xxxxxxxxxxxx]

PARAMETER:

1) 12 character ASCII string.

PURPOSE:

This command causes the client to read the specified GENERIC ACCESS TEST **script**. The **script** is a sub-program that executes a series of commands that measure ACCESS TIME in the sequence dictated by the script.

SCRIPT FILE:

The script is a plain ASCII text file, which contains the following rules and syntax:

1) All numeric values must be hexadecimal.

2) All entry values must be delimited with commas ','.

3) Comments can be made using the ';' semicolon character only.

4) You can comment out an entry by placing the semicolon at the first column only.

5) The script must end with the uppercase keyword **END** in the last line of the script.

The script is entries are defined in the following table:

Entry Field	Max Chars	Description
1	80	User string. ASCII text 80 chars max.
2	8	Starting LBA (-1 will denote 1st LBA on the device).
3	8	Ending LBA (-1 will denote last LBA on the device).
4	4	Number of seeks (number of passes on Seq and crescendo tests).
5	8	Seek increments. Distance between access. (Ignored in modes 2 & 3).
6	2	MODE (bits 0-3) and FLAG (bits 4-7) Modes: 0 - Sequential forward. 1 - Sequential reverse. 2 - Repetitive. 3 - Random. 4 - Crescendo (butterfly). FLAGS: Unused.
7	4	Abort limit MIN (units of 100 us). Ignore if zero.
8	4	Abort limit average MIN (units of 100 us). Ignore if zero.
9	4	Abort limit average MAX (units of 100 us). Ignore if zero.
10	4	Abort limit MAX (units of 100 us). Ignore if zero.

SAMPLE SCRIPT:

```
;*************************************************************
;* Filename: quick.gat
;* Desciption: Sample Generic Access Test script file.
;*************************************************************
"Seq Fwd Seek",0,1000,1,100,00,00,00,00,00
"Seq Rev Seek",0,1000,1,100,01,00,00,00,00
"Rep Test 00 to 1000",0,1000,80,00,02,00,00,00,00
"Rndm Seek Test",-1,-1,100,00,03,00,00,00,00
"Crescendo Seek",0,1000,1,100,04,00,00,00,00

END
```

RESULTS: (displayed in report file)

TEST NAME	COUNT	MIN(msec)	MAX(msec)	AVG(msec)
Seq Fwd Seek	1	18.10	109.00	125.90
Seq Rev Seek	1	8.10	122.10	79.00
Rep Test 00 to 1000	128	2.30	301.60	215.10
Rndm Seek Test	256	39.20	571.00	257.20
Crescendo Seek	1	86.10	560.10	175.50

5. OPEN / CLOSE TRAY itrs=[xxxx] op=[y]

PARAMETER:

1) 4 digit numeric value.

2) 1 digit numeric value.

PURPOSE:

This command provides a method to exercise the media tray mechanism by issuing the ATAPI START/STOP command. Each command issued will alternate the setting of the START bit in the packet thus causing the device to open or close it's tray.

The **itrs** argument specifies the number of START/STOP commands to be issued by this macro command.

The **op** argument sets the initial condition of the START bit in the packet.

EXAMPLE:

OPEN/CLOSE TRAY itrs=10 op=0 // Close tray first, then repeat 9x

6. DASP / PDIAG TIMING

Not implemented in Orion

7. TEST CD TRANSFER RATE SUITE A

Not implemented in Orion

8. TEST CD ACCESS TIME SUITE A

Not implemented in Orion

6.2.1.2 Loop & Utility Commands

Chain to Program

Clear Defect Buffer

End Loop #1 }

End Loop #2 }

End Loop #3 }

End Timed Loop

Force Test Failure

LED's Off

LED <green> ON

LED <red> ON

Print Test Summary

Return from Chained Pgm

Start Loop #1 {

Start Loop #2 {

Start Loop #3 {

Timed Loop (secs)

Update Image File

Wait Delay (*0.1 sec)

1. CHAIN TO PROGRAM name=XXXXXXXX

PARAMETER:

8 character file name string

PURPOSE:

This command allows the download and execution of a new script file. All variables, error counters, defect maps, etc. are global variables and are passed unchanged to the new script. If the new script program modifies any variables, the modified variables will be returned to the calling script. Script programs may be chained up to 32 levels deep. This allows modular program development, mid-test abort limit changes, and greater flexibility. See also *RETURN FROM CHAINED PROGRAM*.

2. CLEAR DEFECT BUFFER

PARAMETER:

none

PURPOSE:

When this command is issued, the test port will erase all logged defects from the defect buffer and clear the defect counters. In addition, the abort limit counters will be reset to zero.

3. } END LOOP #1

PARAMETER:

none

PURPOSE:

This command terminates a START LOOP #1 command. If the loop count set by the START LOOP #1 command is not yet zero, execution will continue at the line following the START LOOP #1 command. If the loop count has decremented to zero, the next script line beyond the END LOOP #1 will be executed.

4. } END LOOP #2

PARAMETER:

none

PURPOSE:

This command terminates a START LOOP #2 command. If the loop count set by the START LOOP #2 command is not yet zero, execution will continue at the line following the START LOOP #2 command. If the loop count has decremented to zero, the next script line beyond the END LOOP #2 will be executed.

5. } END LOOP #3

PARAMETER:

none

PURPOSE:

This command terminates a START LOOP #3 command. If the loop count set by the START LOOP #3 command is not yet zero, execution will continue at the line following the START LOOP #3 command. If the loop count has decremented to zero, the next script line beyond the END LOOP #3 will be executed.

6. } END TIMED LOOP

PARAMETER:

none

PURPOSE:

This command terminates a TIMED LOOP command. If the time set by the TIMED LOOP command has **not** expired, execution will continue at the line following the TIMED LOOP command. If the time has expired, the next script line beyond the END TIMED LOOP will be executed.

7. FORCE TEST FAILURE = xxxxxxxx

PARAMETER:

0 to 8 character alphanumeric string

PURPOSE:

This command causes an immediate abort of the current running test script. This command is useful where a 'script determined' failure has occurred. This command can also be used in an ON ERROR GOSUB construct to force a test failure.

The 8-character string is inserted in the IMAGE file and is also printed on the report output.

8. TURN BOTH LEDS OFF

PARAMETER:

none

PURPOSE:

Turns all port display LED's off.

9. LED GREEN ON

PARAMETER:

none

PURPOSE:

This command turns **on** the GREEN display LED.

10. LED RED ON

PARAMETER:

none

PURPOSE:

This command turns **on** the RED display LED.

11. PRINT TEST SUMMARY

PARAMETER:

none

PURPOSE:

This command forces the SERVER to generate a new binary IMAGE file and an ASCII test report file. The image file name is created from the physical RACK and PORT number (RxxPxxDx.IMG). The report file name is created from the first eight characters of the device serial number (if valid) or alternatively, uses the naming convention of the IMAGE file (RxxPxxDx.REP).

12. RETURN FROM CHAINED PROGRAM

PARAMETER:

none

PURPOSE:

Returns from a chained program to the 'parent' program from which it had chained. Chain nesting with returns may be up to 32 levels deep. Each RETURN statement returns to the previous caller. See also *CHAIN TO PROGRAM*.

Note that any variables that have been modified in a chain program are ‘global’ and remain modified after the chain returns.

13. START LOOP #1 pass=XXXXX

PARAMETER:

A 5 digit decimal value to set the number of loops to perform (0 - 65535).

PURPOSE:

START LOOP #1 forms a basic LOOP construct and follows the same rules as any programming language. Loops can be nested up to 3 deep by using the START and END LOOP 1,2, and 3 statements.

Note that a loop count of one will execute the script twice. A loop count of **ZERO** will execute the script inside the loop structure only once (no looping).

EXAMPLE:

```
** Simple Loop Example – executes <command> 100 times

{ START LOOP #1 pass=99

<command>

<command>

} END LOOP #1


{ START LOOP #1 pass=10     ** Nested Loop Example (66 loops)

{ START LOOP #2 pass=5

<command>

} END LOOP #2

} END LOOP #1
```

14. START LOOP #2 pass=XXXXX

PARAMETER:

A 5 digit decimal value to set the number of loops to perform (0 - 65535).

PURPOSE:

See START LOOP #1.

15. START LOOP #3 pass=XXXXX

PARAMETER:

A 5 digit decimal value to set the number of loops to perform (0 - 65535).

PURPOSE:

See START LOOP #1.

16. TIMED LOOP sec=XXXXX

PARAMETER:

A 5 digit decimal value to set the number of seconds that the loop will perform (0 - 65535).

PURPOSE:

Timed loops allow script instructions to be performed until the specified number of seconds has elapsed.

EXAMPLE:

```
** Simple Loop Example – executes <command> for 1 minute

{ TIMED LOOP sec=60

<command>

<command>

} END TIMED LOOP
```

NOTE: Variable VO bit 21 can be used to escape macro commands inside of a timed loop. The following example shows a program sequence that escapes a timed loop even if the script is executing a MACRO type command:

```
001 SET VARIABLE v=VO (x)n=00200000   // bit 21 (abort macros)

002 { Timed Loop sec=60         // loop for 60 seconds

003 WRITE INCR FWD pass=3       // MACRO takes a long time but -

004 } End Timed Loop            // cmd will be aborted in 60 sec.
```

The VO bit 21 will force escape of any MACRO commands when the timed loop expires. Note that setting the VO bit 21 will force the MACRO command to escape but not the timed loop. If there are more commands inside the timed loop, the next command after the MACRO will be executed.

17. UPDATE IMAGE FILE

PARAMETER:

none

PURPOSE:

Causes the IMAGE file to be updated (refreshed). NOTE: This function is not currently supported.

18. WAIT DELAY (x .1 sec)=XXX

PARAMETER:

3 digit decimal value in the range of 0 - 255.

PURPOSE:

This command is used to insert a delay into the script program. It can be used anywhere in the script program. The value entered is multiplied by 0.1 seconds. As an example, an entry of 50 creates a 5 second delay. Delays therefore can range from 0 to 25.5 seconds of delay. For longer delays, use the TIMED LOOP command or place the WAIT DELAY command inside of a LOOP structure.

6.2.1.3 Oven Control Commands

Run Oven Program

Wait for Oven Step

Wait for Oven Temperature

1. RUN OVEN PROGRAM prog=[xxxx] line=[yy]

PARAMETER:

1) 4 digit decimal value (0 - 9999).

2) 2 digit decimal value (0 - 99).

PURPOSE:

This command is used in a P25 or P2E script to start an oven program or to advance a running oven program to another point in its profile (oven MASTERING). The first argument specifies the program number to run (index value into the oven program table in the setup menu). The second argument specifies the line number (usually 1) to begin executing.

Other arguments that may be used in the ***prog*** field are:

BEG - Start the selected oven program.

END - Stop the current oven program. Also returns the oven to the STANDBY set points.

INC,+ - Increment (advance) the current oven program to the next line. This causes the oven to synchronize to the Orion test script if all test ports have reached a consensus.

If more than one test port is running a script containing this statement, a list is generated to keep track of the ports that are mastering the oven. All of those ports must reach the same point in their respective scripts before any action is taken (port synchronization). The ports, in this mode, are mastering the oven operation and must be in sync to modify the oven profile (batch mode). The exceptions to this rule are as follows:

1) A port passes or fails it's test during operation (port de-registers from the mastering list).

2) A port fails to respond to the SERVER within a timeout period. This port is then excluded from the mastering list. This timeout is set in the system setup menu under the Oven tab (Group Timeout).

2. WAIT FOR OVEN STEP=[xx]

PARAMETER:

1) 2 digit decimal value (0 - 99).

PURPOSE:

This command is used to SLAVE the test port script to the specified oven program line number. When this command is executed in the script, the test port will wait until the oven test program has reached the specified line number before continuing. This command is used to synchronize the test ports with the oven profile.

3. WAIT FOR OVEN TEMPERATURE=[xxxx]

Not implemented.

6.2.1.4 IDE Commands

Configure Port

Disable Drive Retries

Download Microcode

Initiate Diagnostics

Initialize Drive Parameters

Load Zone Table

Read Register

Read Task File Regs

Reset Drive

Select ATA Mode

Set Data Mode

Set Host DMA Mode

Set Feature Register

Set I/O Timing

Set Multiple Mode

Set Sector Overlay

Set Slave Mode

Set Xfer Length

Vendor Unique Command

Vendor Unique Command (48 bit)

Write Register

1. CONFIGURE PORT {IDE}

PARAMETER:

none

PURPOSE:

This command retrieves geometry and configuration data from the device through the IDENTIFY DRIVE ATA command and sets the test port to operate within those parameters. The command also initializes the drive. This command should be issued after the drive power has been enabled either through the TEST SPIN UP command or by issuing a DRIVE POWER ON command.

The CONFIGURE PORT derives the serial number and model number of the drive. The eight least significant characters of the serial number are used to generate the REPORT output file and the .DAT binary database file.

THEORY OF OPERATION

1) Recalibrate the drive. Wait for ready if necessary.

2) Issue an **IDENTIFY DRIVE** ATA command.

3) Read 1 block of data from the drive.

4) Check the drive status.

5) Set maximum head, maximum cylinder, and number of sectors as derived from the identify data.

6) Reset MIN HEAD and MIN CYL to zero.

7) Issue the **INITIALIZE DRIVE PARAMETERS** ATA command using the above configuration data.

SEE ALSO:

CONFIGURE PORT (SCSI) 6.2.1.8

CONFIGURE PORT (ATAPI) 6.2.1.1

2. DISABLE DRIVE RETRIES (disable=1)

PARAMETER:

1 digit numeric value (0 - 9). **Default = 0**.

PURPOSE:

This command allows the user to disable drive retries by entering a '0' or '1' argument (T/F). This command changes the LSB of the IDE read and write commands to enable or disable retries. The drive, depending on vendor firmware implementation, may disable internal retries. The commands are summarized as follows:

```
Flag      R/WCommand Op Code

0         Read   20 hex          // Enables retries

1         Read   21 hex          // disables retries
```

```
0        Write 30 hex          // Enables retries

1        Write 31 hex          // disables retries
```

3. DOWNLOAD MICROCODE

PARAMETER:

none

PURPOSE:

This macro command automatically downloads Microcode firmware to the device using the devices DOWNLOAD FIRMWARE command. The source file name must have been set using the *FILENAME* command. This command will work for IDE, ATAP

NOTE: This command is not currently supported.

4. INITIATE DIAGNOSTICS

PARAMETER:

none

PURPOSE:

This command issues the ATA *Initiate Diagnostics* command to the device. Results are checked and a failure causes the test to abort.

5. INITIALIZE DRIVE PARAMETERS (ATA)

PARAMETER:

none

PURPOSE:

This command uses the values derived by the CONFIGURE PORT command to initialize the drive parameters. The values can also be overridden by the SET MAX HEAD and SET MAX CYLINDER commands. Use this command to change the logical configuration of the drive.

6. LOAD ZONE TABLE

This command is present for legacy compatibility but is not supported in Orion.

7. READ REGISTER adr=[xx] var=[yy]

PARAMETER:

1) 2 digit numeric value of the I/O address of the task register to read (0 - 99).

2) 2 character ASCII string that defines the destination variable.

PURPOSE:

This command allows the user to read any readable task file register into the specified variable. The first argument is the I/O offset of the register to be read. The second argument is the destination variable.

EXAMPLE:

READ REGISTER adr=07 var=B0 // reads the ATA status register into B0

The following table shows register access versus address:

Address	ATA Register
00	Data Register
01	Error Register
02	Sector Count Register
52	Sector Count Register (15:8) 48 bit mode only
03	Sector Number Register
53	Sector Number Register (LBA 31:24) 48 bit mode only
04	Cylinder LOW Register
54	Cylinder LOW Register (LBA 39:32) 48 bit mode only
05	Cylinder HIGH Register
55	Cylinder HIGH Register (LBA 47:40) 48 bit mode only
06	Drive/Head Register

07	Status Register
08	Data Register EVEN (LSB of data word)
09	Data Register ODD (MSB of data word)
0E	Alternate Status register

For more information on ATA taskfile registers, see the ATA specification.

8. READ TASK FILE REGS

PARAMETER:

none

PURPOSE:

Reads all readable ATA task file registers into specific variables:

YE - Error register

YC - Sector count register

YN - Sector number register

YL - Cylinder low register

YH - Cylinder high register

YD - Drive/head register

YA - Alternate status register

YS - Status register

YB - Drive address register

9. RESET DRIVE

PARAMETER:

none

PURPOSE:

Issues a 20 millisecond hard reset to pin 1 of the ATA interface. Since this resets the device, the device will be placed in its default configuration.

10. SELECT ATA mode=[x]

PARAMETER:

1 digit numeric value (0 - 9). **Default = 0.**

PURPOSE:

In the Legacy system, this command defines the drive interface mode as ATA. In the Orion system, the argument has the following meaning:

0 = CHS (physical mode)

2 = LBA (28 bit) (logical mode standard)

3 = LBA (48 bit) (logical mode extended)

11. SET DATA MODE #=[x]

PARAMETER:

1 digit numeric value (0 - 9). **Default = 0.**

PURPOSE:

Allows the use of various pre-defined data patterns.

0 - User defined (set by the SET DATA PATTERN commands). If no patterns are specified, the client will choose a default data pattern.

1 - Random data (no seed).

2 - Random data (LBA seed)

3 - Incrementing data pattern.

4 - Decrementing data pattern.

5 - Walking 1's data pattern (0x80, 0x40, 0x20, etc...).

6 - Walking 0's data pattern (0x7F, 0xBF, 0xDF, etc...).

Values higher than 6 are ignored.

12. SET HOST DMA mode=[x]

PARAMETER:

1 digit numeric value (0 - 9). **Default = 0.**

PURPOSE:

This command sets up the DMA transfer mode for the ATA device under test by sending a SET FEATURES command to the DUT.

0 - Disables DMA mode (default to PIO mode).

1 – N.A.

2 - Multiword DMA mode (see SET MEMORY I/O TIMING)

3 – N.A.

4 - UDMA mode 0 (16.6 Mb/S)

5 - UDMA mode 1 (25 Mb/S)

6 - UDMA mode 2 (33 Mb/S)

7 – UDMA mode 3 (50 Mb/S)

8 – UDMA mode 4 (66 Mb/S)

9 – UDMA mode 5 (100 Mb/S)

10 – UDMA mode 6 (133 Mb/S)

13. SET FEATURES REGISTER byte=[xx]

PARAMETER:

2 digit hexadecimal value (0 – 0xFF).

PURPOSE:

Writes the indicated value to the ATA FEATURES register. See ATA specification for functions.

14. SET MEM / IO TIMING p/w=[xxx] spc=[yyy]

PARAMETER:

1) 3 digit numeric value (0 - 255) for Pulse Width. **Default = 5**.

2) 3 digit numeric value (0 - 255) for Spacing. **Default = 15**.

PURPOSE:

This command controls the pulse width and spacing of IORD and IOWR for PIO and Multiword DMA modes. The timings are set to a resolution of 30 nanoseconds.

Various timings are set as follows:

> The following are settings to replicate various PIO and DMA mode timings:

	P/W	SPACING	Cycle(ns)	Mb/s
PIO mode 0 -	5	15	600	3.33
PIO mode 1 -	3	9	400	5
PIO mode 2 -	2	4	250	8
PIO mode 3 -	1	2	175	11.4
PIO mode 4 -	1	0	125	16
PIO 20 mb/s -	0	0	100	20
MW DMA 0 -	7	8	475	4.2
MW DMA 1 -	1	1	150	13.33
MW DMA 2 -	1	0	125	16

Note: These settings have no effect in UDMA modes.

15. SET MULTIPLE MODE secs/blk=[xxx]

PARAMETER:

3 digit numeric value (0 - 255). **Default = 0** (disabled).

PURPOSE:

This command allows the user to enter the number of sectors to transfer during a MULTIPLE MODE data transfer.

If the argument is set to 0, multiple mode is disabled.

If the argument is set to 255, the default value is used, which is read from the drive during CONFIGURE PORT. Otherwise the argument will override the block count.

16. SET SECTOR Overlay=[x]

PARAMETER:

A 1 digit numeric value (0 - 9). **Default = 3**.

PURPOSE:

This command is used to force a sector unique address into each block written to the drive. The overlay is a 4-byte data field, which is the LBA of the drive and can be placed at the beginning of the block, end of the block, or both as follows:

0 - Sector overlay at the START and END of the block.

1 - Sector overlay at the START of the block.

2 - Sector overlay at the END of the block.

3 - Disable sector overlay (**default**).

17. SET SLAVE mode=[x] csel=[y]

PARAMETER:

1) 1 digit numeric value (0 - 9). **Default = 0** (master mode).

2) 1 digit numeric value (0 - 9).

PURPOSE:

This command controls the addressing of an IDE MASTER or SLAVE device. If **mode**=0, the IDE device is addressed as a MASTER (default), whereas if **mode**=1, the IDE device is addressed as a SLAVE.

NOTE: The **csel** argument is ignored since Orion hardware has no support for CABLE SELECT.

18. SET XFER LENGTH (#secs)=[xxx]

PARAMETER:

3 digit numeric value (0 - 255). **Default = 0**.

PURPOSE:

This command allows the user to set the number of sectors to transfer during READ or WRITE operations. An argument of 0 will yield a transfer length of 256 sectors. This is the ***default*** and is the maximum number of sectors that can be transferred during one command operation by the ATA interface.

19. VENDOR UNIQUE, [xx,xx,xx,xx,xx,xx,xx,T]

PARAMETER:

Up to seven - 2 digit hexadecimal characters and a single ASCII Terminator character.

PURPOSE:

The VENDOR UNIQUE command allows direct access to the main ATA task file registers. In order, from left to right, they are:

COMMAND, DRV/HEAD, CYL HIGH, CYL LOW, SEC COUNT, SEC NUMBER, and optionally FEATURES.

All registers must contain some value (except the FEATURES register which is optional). All registers except the command register may be substituted by variables. Valid variables for use in the VU string are as follows:

P1, N2, S1, X1 (see example).

The vendor unique string may end with a special *COMMAND TERMINATOR* character, which defines the expected data transfer direction of the command. If no terminator is present, the command is not expected to transfer any data.

EXAMPLE 1:

The following script examples show the general usage of the VEN UNQ command.

```
01 * SEND VU CMD TO DUT

02 TEST SPIN UP

03 CONFIGURE PORT

04 VEN UNQ, 30,A0,00,00,01,01,O   (write one sector from track
buffer)
```

EXAMPLE 2:

```
01 *  RANDOM TEST USING VEN UNQ COMMAND

02 *
```

```
03 *

04    TEST SPIN UP

05    CONFIGURE PORT

06 *  INSURE THAT THE DATA PATTERN IS SET

07    WRITE CURRENT CYL

08    READ CURRENT CYL

09  { Start Loop #1 loops=65534

10    SET RANDOM VARIABLE v = P1

11    SET RANDOM VARIABLE v = N0

12    SET RANDOM VARIABLE v = S1

13    SET RANDOM VARIABLE v = X1

// write random cyl, head, sector, xfer len

14    VEN UNQ,31,P1,N2,N1,X1,S1,O

15    VEN UNQ,21,P1,N2,N1,X1,S1,I     // read it back

16  } End Loop #1
```

20. VENDOR UNIQUE (48 bit), [xx,xx,xx,xx,xx,xx,xx,T]

PARAMETER:

Up to twelve - 2 digit hexadecimal characters and a single ASCII Terminator character.

PURPOSE:

The VENDOR UNIQUE command allows direct access to the main ATA task file registers. In order, from left to right, they are:

COMMAND,
DRV/HEAD,
LBA (47:40)
LBA (39:32)
LBA (31:24)
LBA (23:16)
LBA (15:8)
LBA (7:0)
SEC COUNT Ext
SEC COUNT

FEATURE Ext (Optional)
FEATURE (Optional)
All registers must contain some value (except the FEATURE registers which are optional).

The vendor unique string may end with a special *COMMAND TERMINATOR* character, which defines the expected data transfer direction of the command. If no terminator is present, the command is not expected to transfer any data.

21. WRITE REGISTER adr=[xx] var=[yy]

PARAMETER:

1) 2 digit numeric value (0 - 99).

2) 2 digit string

PURPOSE:

This command allows the user to write any write-able ATA task file register from the specified variable. The **adr** variable defines the I/O offset address of the task file register and **var** is the variable from which the data is written.

Address	ATA Register
00	Data Register
01	Features Register
02	Sector Count Register
52	Sector Count Register (15:8) 48 bit mode only
03	Sector Number Register
53	Sector Number Register (LBA 31:24) 48 bit mode only
04	Cylinder LOW Register
54	Cylinder LOW Register (LBA 39:32) 48 bit mode only
05	Cylinder HIGH Register
55	Cylinder HIGH Register (LBA 47:40) 48 bit mode only
06	Drive/Head Register
07	Command Register

08	Data Register EVEN (LSB of data word)
09	Data Register ODD (MSB of data word)
0E	Device Control Register

6.2.1.5 Program Control Commands

Add Defect To List

Add to Variable

Add Var to Var

And Variables

Call Subroutine

Comment

Compare Buffers

Compare Variables

Copy Buffer to Buffer

Custom Spawn

Divide Var by Var

Download File

Filename (upld/dnld)

Goto <label>

IF <cc> THEN NEXT INSTR

Inc Hard Errors

Inc Soft Errors

Label:

Load New Abort Limits

Load Var from Buffer

Mov Var to Var

Multiply Var by Var

On Error Gosub

Or Variables

Pop Var

Push Var

Return from Subroutine

Return from On Error Gosub

Select Buffer

Set Random Variable

Set Variable

Store Var to Buffer

Subtract from Var

Subtract Var from Var

Upload File

Xor Variables

1. ADD DEFECT TO LIST

PARAMETER:

none

PURPOSE:

Adds the current LBA to the defect list. This command can be used in an ON ERROR GOSUB subroutine to add errors to the list.

2. ADD TO VARIABLE v=[xx] (x)n=[yyyy]

PARAMETER:

1) 2 digit ASCII string.

2) 4 digit hexadecimal value (0 - FFFF).

PURPOSE:

Issue this command to **add** the immediate value [yyyy] to the specified variable [xx].

EXAMPLE:

```
ADD TO VARIABLE v=A0  (x)n=FF00    // A0 <- A0 + 0xFF00
```

3. ADD VAR s=[xx] to VAR d=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

This command **adds** the two specified variables together and places the sum in the **destination** variable.

4. AND VARIABLES d=[xx] s=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

This command performs a logical bitwise **and** on the two specified variables and places the result in the (D)estination variable.

5. CALL SUBROUTINE :[xxxxxx]

PARAMETER:

1) 6 digit ASCII string.

PURPOSE:

The CALL command jumps to the label [xxxxxx] and saves a pointer to the line following the CALL command so that a subsequent *RETURN FROM SUBROUTINE* command can resume operation.

EXAMPLE:

```
01 *  CALL SUBR EXAMPLE

02    CALL MYSUBR

03    GOTO :PGMEND

04 MYSUBR:

05 * DO WHATEVER...

06    RETURN FROM SUBROUTINE

07 *

08 PGMEND:
```

*6. COMMENT * [xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]*

PARAMETER:

1) 30 character ASCII string.

PURPOSE:

Use the COMMENT command to document script programs. Comments are embedded in the script but are not executed.

7. COMPARE BUFFERS s=[xx] d=[yy]

PARAMETER:

1) 2 character ASCII string.

2) 2 character ASCII string.

PURPOSE:

Use this command to compare two buffers. The result of the compare sets the condition code used with the “IF <cc> THEN NEXT INSTR” command to determine program flow. The condition code result from this command can be:

E - Equal

N - Not equal

< - Less than

> - Greater than

Note that the variable RT will contain the location of any miscompare that is discovered.

8. COMPARE VARIABLE s=[xx] d=[yy]

PARAMETER:

1) 2 character ASCII string.

2) 2 character ASCII string.

PURPOSE:

Use this command to compare two variables. The result of the compare sets the condition code used with the “IF <cc> THEN NEXT INSTR” command to determine program flow. The condition code result from this command can be:

E - Equal

N - Not equal

< - Less than

> - Greater than

9. COPY BUFFER TO BUFFER src=[x] dest=[y]

PARAMETER:

1) 1 character ASCII string.

2) 1 character ASCII string.

PURPOSE:

Copies any of the system buffers to any other buffer from 'SRC' to 'DEST'. This command accepts two single character arguments to define the source and destination buffers as follows:

T - Track Buffer (same as Read buffer)

0 - Scratch Buffer 0 (64K)

1 - Scratch Buffer 1 (64K)

2 - Scratch Buffer 2 (64K)

3 - Scratch Buffer 3 (64K)

R – Read Buffer (512K)

W – Write Buffer (512K)

10. CUSTOM SPAWN spawn=[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]

PARAMETER:

1) 30 character ASCII string.

PURPOSE:

This function allows the user to call an external process (Executable or Batch file) located on the **SERVER** system. When the external process is completed, control returns to the script. The 30-character string defines the name of the process and any command line arguments that may be required. Due to legacy compatibility, the filename must be limited to eight characters.

11. DIV VAR s=[xx] to VAR d=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

This command **divides** the two specified variables together and places the result of the division back into the **destination** variable. See variable RM for remainder.

12. DOWNLOAD FILE

PARAMETER:

none

PURPOSE:

This command is used to DOWNLOAD a file from the Server to the Client and into the currently selected buffer. The file can be of any type but may not exceed the size of the destination buffer (up to 512 KB). The command assumes that the script has set the requested filename via the *FILENAME* command.

See also variable DL which can be set to control the length of the file download into the buffer.

13. FILENAME = [xxxxxxxxxxxx]

PARAMETER:

1) 12 character ASCII string.

PURPOSE:

This command sets the filename for subsequent UPLOAD or DOWNLOAD FILE commands. It is used to inform the Server of which file is to be moved to/from the Client.

Due to legacy compatibility, the filename is limited to eight characters and the file extension is limited to three characters.

EXAMPLE:

```
01 * TEST WITH +/-15% SERVO OFFSET
```

```
02    FILENAME =PLUS15.OFF

03    TEST SPIN UP

04    CONFIGURE PORT

// vendor unq sets offset to the DUT

05    VEN_UNQ,D0,00,02,F8,00,00,Z

06    WRITE INCR FORWARD pass=1

07    FILENAME =MINUS15.OFF

   // vendor unq sets offset to the DUT

08    VEN_UNQ,D0,00,02,F8,00,00,Z

09    READ INCR REVERSE pass=1
```

14. GOTO :[xxxxxx]

PARAMETER:

1) 6 character ASCII string.

PURPOSE:

Issue this command to perform an unconditional jump to a specified label. The label specifier can be up to 6 characters in length. See also LABEL command.

15. IF cc=[x] THEN NEXT INSTR

PARAMETER:

1) 1 digit ASCII character.

PURPOSE:

The '**IF <cc> THEN NEXT INSTR**' command is used to modify program flow based on the condition code set by a previous compare or similar operation. **IF** the condition code state matches the 'cc' argument, **THEN** the next instruction will be executed, **ELSE** the script will skip to the second instruction following this command. Valid condition code arguments are as follows:

E - Equal.

N - NOT Equal.

> - Greater than.

< - Less than.

EXAMPLE:

```
01    SET VARIABLE var=A0  n(x)=00000001

02    SET VARIABLE var=B0  n(x)=00000000

03    SET VARIABLE var=C0  n(x)=00000000

03    COMPARE VARIABLE s=A0  d=B0

04    IF cc=E THEN NEXT INSTR

05    ADD TO VARIABLE v=C0  n(x)=00000001   // never gets here

06    ADD TO VARIABLE v=C0  n(x)=00000001   //always gets here

07 * EXPECT C0 == 00000001
```

16. INC HARD ERRORS

PARAMETER:

none

PURPOSE:

Increments the Client HARD error counter. This command is useful inside of an ON ERROR GOSUB command to artificially add errors to the error list.

17. INC SOFT ERRORS

PARAMETER:

none

PURPOSE:

Increments the Client SOFT error counter. This command is useful inside of an ON ERROR GOSUB command to artificially add errors to the error list.

18. LABEL :[xxxxxx]

PARAMETER:

1) 6 character ASCII string.

PURPOSE:

Provides a label in the script to allow a target address for GOTO and CALL functions. The label may be up to 6 characters in length and can contain any combination of printable ASCII characters.

19. LOAD NEW ABORT LIMITS file=[xxxxxxxx]

PARAMETER:

1) 8 character ASCII string.

PURPOSE:

Loads the abort limits section of any P25 script file into the current running program. Following this command, the script will function with the new abort limits. This command is useful for changing the abort limits at run time.

20. LOAD VAR FROM BUF v=[xx] disp=[yy]

PARAMETER:

1) 2 character ASCII string.

2) 2 character ASCII string.

PURPOSE:

Issue this command to load the specified 4-byte variable [xx] with data from the currently selected buffer at an offset specified by the variable [yy]. See also *SELECT BUFFER* command.

EXAMPLE:

```
01    SET VARIABLE var=D0 n(x)=00000005

02    LOAD VAR FROM BUF v=A0 disp=D0
```

Buffer = 00000000001020304000000...

Result: Variable A0 = 01020304

21. MOVE VAR s=[xx] TO VAR d=[yy]

PARAMETER:

1) 2 character ASCII string.

2) 2 character ASCII string.

PURPOSE:

Issue this command to move the data content of one variable into another. The destination variable will contain the data of the source variable. The length of the data moved is determined by the size of the destination variable.

22. MULTIPLY VAR s=[xx] BY VAR d=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

This command **multiplies** the two specified variables together and places the result of the multiplication back into the **destination** variable.

23. ON ERROR GOSUB label=[xxxxxx] opt=[yyyy]

PARAMETER:

1) 6 digit ASCII string.

2) 4 digit hexadecimal value (0 - FFFF).

PURPOSE:

Allows the user to define a script based error handler subroutine. The error handler subroutine is constructed the same way as a normal subroutine except that the end statement must be a *RETURN FROM ON ERROR GOSUB*.

The **opt** argument allows various options to control the behavior of the ON ERROR GOSUB function. For more information see appendix F.

NOTE: If the program that contains a ON ERROR GOSUB command chains to another program, the ON ERROR GOSUB error handler will be lost. In this case, it is necessary to establish an ON ERROR GOSUB handler inside the chained module.

24. OR VARIABLES d=[xx] s=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

Performs a logical bitwise OR between the two specified variables and places the result in the (D)estination variable.

25. POP VAR v=[xx]

PARAMETER:

1) 2 character ASCII string.

PURPOSE:

Issue this command to POP a variable from the variable STACK. Data must have first been PUSHED onto the stack. If the stack is empty, a STACK OVERFLOW error will occur. See also PUSH VAR.

26. PUSH VAR v=[xx]

PARAMETER:

1) 2 character ASCII string.

PURPOSE:

Issue this command to PUSH a variable onto the variable STACK. The STACK is used as temporary storage for variables. Use POP VAR to retrieve the data from the stack. The maximum depth of the variable stack is 32 - 4 byte variables.

27. RETURN FROM SUBROUTINE

PARAMETER:

none

PURPOSE:

Issue this command to RETURN from a subroutine. When RETURN is issued, script execution continues from the next instruction following the *CALL*: [xxxxxx].

28. RETURN FROM ON ERROR GOSUB opt=[xxxx]

PARAMETER:

1) 4 digit hexadecimal value (0 - FFFF).

PURPOSE:

This command terminates an *ON ERROR GOSUB* function. The "**opt**" argument provides the following flag options that affect the exit behavior.

0 = Disable entry into the 'normal' error handler.

1 = Enable entry into the 'normal' error handler.

29. SELECT BUFFER b=[x]

PARAMETER:

1) 1 digit ASCII character. **Default = 'T'**.

PURPOSE:

This command selects a specific buffer for upload / download operations. The valid arguments are as follows:

R - Read buffer.

T - Track buffer (same as write buffer).

W - Write buffer.

0 - Scratch buffer 0.

1 - Scratch buffer 1.

2 - Scratch buffer 2.

3 - Scratch buffer 3.

30. SET RANDOM VARIABLE v=[xx]

PARAMETER:

1) 2 digit ASCII string.

PURPOSE:

Issue this command to initialize the specified 4-byte variable with a random number. Since the random number generator is pseudo random, the random numbers will have the same starting sequence during any script execution. The following variables can be used with this command: L0, N0, P0, R0, S0, X0.

31. SET VARIABLE v=[xx] (x)n=[yyyyyyyy]

PARAMETER:

1) 2 digit ASCII string.

2) 8 digit hexadecimal value (0 - FFFFFFFF).

PURPOSE:

This command is used to set a hexadecimal numeric value [yyyyyyyy] into the specified variable [xx]. See also the description for *VARIABLES*.

32. STORE VAR TO BUFFER v=[xx] disp=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

Issue this command to store the specified variable [xx] into the currently selected buffer at an offset specified in the variable [yy]. See *SELECT BUFFER* command.

EXAMPLE:

```
01    SET VARIABLE v=A0 (x)n=01020304
```

```
02    SET VARIABLE v=D0 (x)n=00000005

03    STORE VAR TO BUF v=A0 disp=D0
```

Result: BUFFER =0000000000010203040000...

33. SUBTRACT FROM VAR v=[xx] (x)n=[yyyy]

PARAMETER:

1) 2 digit ASCII string.

2) 4 digit hexadecimal numeric value (0 - FFFF).

PURPOSE:

This command is used to subtract an immediate value [yyyy] from the specified variable [xx]. The result is stored back into the specified variable [xx].

34. SUBTRACT VAR s=[xx] FROM VAR d=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

This command is used to subtract the source variable [xx] from the destination variable [yy]. The result of the subtraction is placed in the **(d)estination** variable [yy].

35. UPLOAD FILE

PARAMETER:

none.

PURPOSE:

This command will cause data from the currently selected buffer to be uploaded to the SERVER. The filename for the destination file is set via the *FILENAME* command.

See also variable DL to control the number of bytes to upload.

XOR VARIABLE d=[xx] s=[yy]

PARAMETER:

1) 2 digit ASCII string.

2) 2 digit ASCII string.

PURPOSE:

This command is used to exclusive or (XOR) the source variable [xx] with the destination variable [yy]. The result of the XOR is placed in the destination variable [yy].

6.2.1.6 Read Commands

Enable Data Compare

Read Current Track

Read Current Cylinder

Read Increment Fwd

Read Increment Rev

Read Oscillate

Read Random Tracks

Read Random Cylinders

Read Sector

1. ENABLE DATA COMPARE (0=on, 1=off) = [x]

PARAMETER:

1) 1 digit Boolean value (1 or 0: TRUE/FALSE). **Default = 1** (disabled).

PURPOSE:

This command is used to enable or disable the data compare function used during reading. When data compare is enabled, all data read from the device is compared with the expected data.

If a miscompare occurs, the block (sector) and the offset within the block are entered into the error list. Also, the bad data (data read) and the expected data (good data) are saved in the defect list.

2. READ CURRENT TRACK

PARAMETER:

none

PURPOSE:

This command causes the test system to read the currently addressed HEAD of the currently addressed CYL. All sectors on the logical head are read (one media surface). The length of the read is based on the number of sectors per head calculated by the CONFIGURE PORT macro. The transfer length can be overridden using the X0 variable.

If DATA COMPARE is enabled, all data read will be compared against the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

3. READ CURRENT CYL

PARAMETER:

none

PURPOSE:

This command causes the test system to read the currently addressed CYL. All sectors on the logical cylinder are read (one cylinder). The length of the read is based on the number of sectors per cylinder calculated by the CONFIGURE PORT macro. The transfer length can be overridden using the X0 variable.

If DATA COMPARE is enabled, all data read will be compared with the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

4. READ INCR FORWARD pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the test system to read all cylinders in a forward (incrementing) direction from MIN CYL (or MIN LBA) to MAX CYL (or MAX LBA) incremented by the value of the SET SEEK INCREMENT command. The default value for MIN CYL or MIN LBA is zero. The maximum cylinder or LBA is calculated by the CONFIGURE PORT macro. The minimum and maximum values can be set with the SET MIN CYL and SET MAX CYL commands or by setting the variables MN (min) and MX (max). This works both in CHS mode (ATA) or LBA mode (ATA, SCSI, ATAPI, etc.). In LBA mode the values are in logical blocks, in CHS mode the values are in cylinders.

If DATA COMPARE is enabled, all data read will be compared with the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

5. READ INCR REVERSE pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the test system to read all cylinders in a reverse (decrementing) direction from MAX CYL (or MAX LBA) to MIN CYL (or MIN LBA). The default value for MIN CYL is zero. The maximum cylinder or LBA is calculated by the CONFIGURE PORT macro. The minimum and maximum values can be set with the SET MIN CYL and SET MAX CYL commands or by setting the variables MN (min) and MX (max). This works both in CHS mode (ATA) or LBA mode (ATA, SCSI, ATAPI, etc.). In LBA mode the values are in logical blocks, in CHS mode the values are in cylinders.

If DATA COMPARE is enabled, all data read will be compared with the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

6. READ OSCILLATE pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the device to read all sectors on all cylinders while performing an OSCILLATING SEEK. The function is then repeated for the number of passes specified.

If DATA COMPARE is enabled, all data read will be compared against the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

7. READ RANDOM TRACK pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the device to seek to a random cylinder and random head. All sectors are then read on the random head number. The function will repeat for the given number of passes.

The random cylinder and head are limited by MIN/MAX CYL (MN,MX) and MIN/MAX HEAD (HN, HX) values, which are initially set by the *CONFIGURE PORT* command.

If DATA COMPARE is enabled, all data read will be compared with the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

8. READ RANDOM CYLIN pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the device to seek to a random cylinder. All sectors are then read on the random cylinder number. The function will repeat for the given number of passes.

The random cylinder range is limited by MIN CYL (MN) and MAX CYL (MX) values which are initially set by the CONFIGURE PORT command

If DATA COMPARE is enabled, all data read will be compared with the expected data.

If device detected errors occur, retries may be applied according to the SET RETRIES arguments and the type of error encountered.

9. READ SECTOR head=[xx] sec[yyy]

PARAMETER:

1) 2 digit decimal value (0 - 99).

2) 3 digit decimal value (1 - 255).

PURPOSE:

Reads a single sector on the specified head and sector on the current cylinder. The transfer length is automatically set to 1.

If DATA COMPARE is enabled, all data read will be compared with the expected data.

6.2.1.7 Write Commands

Write Current Track

Write Current Cylinder

Write Increment Fwd

Write Oscillating

Write Increment Rev

Write Random Tracks

Write Random Cylinders

Write Sector

1. WRITE CURRENT TRACK

PARAMETER:

none

PURPOSE:

This command causes the test system to write the currently selected HEAD of the currently addressed CYL. All sectors on the logical head are written (one media surface) with the current data pattern. The length of the write is based on the number of sectors per head calculated by the *CONFIGURE PORT* macro. The transfer length can be overridden using the X0 variable.

2. WRITE CURRENT CYL

PARAMETER:

none

PURPOSE:

This command causes the test system to write the currently addressed CYL. All sectors on the logical cylinder are written (one cylinder) with the current data pattern. The length of the write is based on the number of sectors per cylinder calculated by the *CONFIGURE PORT* macro. The transfer length can be overridden using the X0 variable.

3. WRITE INCR FORWARD pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the test system to write all cylinders in a forward (incrementing) direction from MIN CYL (or MIN LBA) to MAX CYL (or MAX LBA) incremented by the value of the SET SEEK INCREMENT command and using the current data pattern. The default value for MIN CYL or MIN LBA is zero. The maximum cylinder or MAX LBA is calculated by the *CONFIGURE PORT* macro. The minimum and maximum values can be set with the SET MIN CYL and SET MAX CYL commands or by setting the variables MN (min) and MX (max). This works both in CHS mode (ATA) or LBA mode (ATA, SCSI, ATAPI, etc.). In LBA mode the values are in logical blocks, in CHS mode the values are in cylinders.

4. OSCILLATING WRITE pass=[xxxxx]

PARAMETER:

1) 5 digit numeric value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the device to write all sectors on all cylinders while performing an OSCILLATING SEEK. The function is then repeated for the number of passes specified.

5. WRITE INCR REVERSE pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the test system to write all cylinders in a reverse (decrementing) direction from MAX CYL (or MAX LBA) to MIN CYL (or MIN LBA) decremented by the value of the SET SEEK INCREMENT command. The default value for MIN CYL is zero. The maximum cylinder or MAX LBA is calculated by the CONFIGURE PORT macro. The minimum and maximum values can be set with the SET MIN CYL and SET MAX CYL commands or by setting the variables MN (min) and MX (max). This works both in CHS mode (ATA) or LBA mode (ATA, SCSI, ATAPI, etc.). In LBA mode the values are in logical blocks, in CHS mode the values are in cylinders.

6. WRITE RANDOM TRACK pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the device to seek to a random cylinder and random head. All sectors are then written on the random head number (one media surface) using the current data pattern. The function will repeat for the given number of passes.

The random cylinder and head are limited by MIN/MAX CYL (MN,MX) and MIN/MAX HEAD (HN,HX) values which are initially set by the *CONFIGURE PORT* command.

7. WRITE RANDOM CYLIN pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). 0 = 65536 iterations.

PURPOSE:

This command causes the device to seek to a random cylinder. All sectors are then written on the random cylinder number using the current data pattern. The function will repeat for the given number of passes.

The random cylinder range is limited by MIN CYL and MAX CYL values which are initially set by the *CONFIGURE PORT* command.

8. WRITE SECTOR head=[xx] sec=[yy]

PARAMETER:

1) 2 digit decimal value (0 - 99).

2) 3 digit decimal value (1 - 255).

PURPOSE:

Writes a single sector on the specified head and sector. The transfer length is automatically set to 1 sector.

6.2.1.8 SCSI Commands

Build SCSI CDB

Cmd Complete T.O. Gosub

Configure Port (SCSI)

Continue on Error

Convert Defect List

End Que

Load Zone Table

Parity Control Mode

Read Defect List

Reassign Blocks Macro

Reset Drive

Set 1-8 Byte Pattern

Set Auto Error Check

Set Identify Msg (TCQ)

Set Initiator ID

Set LUN Address

Set SCSI Mode

Set Sync Op Mode

Set #LBA TO XFER

Set Target ID

Start QUE

Test Parity

Test Sync Data Transfers

1. BUILD SCSI CDB

PARAMETER:

1) 24 digit hexadecimal (comma delimited) string.

PURPOSE:

This command allows the construction of a SCSI Command Descriptor Block (CDB) of up to 12 bytes in length plus an optional *COMMAND TERMINATOR* character.

EXAMPLE:

VEN UNQ 0A,01,02,03,04,05,06,07,08,09,10,11,T

2. IF CMD T.O. Gosub=[xxxxxx] retry=[yy]

PARAMETER:

1) 6 character ASCII string.

2) 2 digit decimal numeric value (0 - 99).

PURPOSE:

NOTE: This command is included for legacy compatibility but is ***NOT*** supported in Orion.

3. CONFIGURE PORT {SCSI}

PARAMETER:

none

PURPOSE:

Issue this command to set the SCSI test port parameters such as MAX LBA, device type, ANSI spec, etc. The following sequence is used by the *CONFIGURE PORT* command:

TEST UNIT READY is issued up to two times to clear any power up CHECK CONDITIONS. If the device reports a CHECK CONDITION, a REQUEST SENSE is issued to clear the error condition.

INQUIRY command is issued to gather information about the device such as model number, serial number, device type, etc.

READ CAPACITY command is issued to read the devices maximum Logical Block Address (LBA). This command is done twice - first to obtain the devices maximum LBA which is placed in the MX variable. The second time to obtain the highest LBA on cylinder zero (blocks per cylinder).

4. CONTINUE ON ERROR (1=enab) =[x]

PARAMETER:

1) 1 digit Boolean value (TRUE/FALSE). **Default = 0** (FALSE).

PURPOSE:

Each time a P25 script is executed, the status of the device is checked. If a fatal SCSI error occurs, and if the CONTINUE ON ERROR is enabled (TRUE), the test will continue to execute.

5. CONVERT DEFECT LIST (0=BFI, 1=Phy, 7=LBA) = [x]

PARAMETER:

1) 1 digit numeric value. Default = 7 (LBA).

PURPOSE:

This command allows the defect list in the buffer to any of the three available formats listed above:

BFI - Bytes from Index (not supported in Orion).

PHY – Convert defect list to physical (CHS) format.

LBA – Convert defect list to logical (LBA) format.

Note: This command only works if the device supports the SCSI-II translate address commands.

6. END QUE

PARAMETER:

8-character file name string

PURPOSE:

This command defines the end of a SCSI *START QUE* block.

7. LOAD ZONE TABLE = [xxxxxxxx]

PARAMETER:

1) 8 character ASCII file name string.

PURPOSE:

This command is **not supported** in the Orion P25 test system. The command will be ignored if included in a script.

8. PARITY CONTROL (0-3) = [x]

PARAMETER:

1) 1 digit numeric value. **Default = 2**.

PURPOSE:

This command enables or disables the Parity mode of the SCSI device depending on the argument value:

0 - Disable parity generation and checking.

1 - Odd parity is tested during a data in phase. No parity is generated on a data out phase.

2 - Odd parity is generated during out phase. Parity is not checked during a data in phase.

3 - Odd parity is checked and generated on both data in and data out phases.

9. READ DEFECT LIST FROM DRV

PARAMETER:

none

PURPOSE:

This command is issued to read the defect list from a SCSI device. The command reads either the primary defect list or the grown defect list depending on the FORMAT TYPE byte. See *SET FORMAT TYPE CODE* command.

10. REASSIGN BLOCKS MACRO

PARAMETER:

none

PURPOSE:

This macro command is intended for SCSI devices that don't support multiple defect entries when using the REASSIGN BLOCK command. This macro will issue a REASSIGN BLOCK command for each entry in the defect list.

11. RESET DRIVE

PARAMETER:

none

PURPOSE:

This command will assert the hard device reset to the SCSI bus. Reset is asserted for approximately 25 Usec., then the client will wait for approximately 1 second before continuing script execution.

12. 1-8 BYTE PATN num=[x] Byte(x)=[yyyyyyyyyyyyyyyy]

PARAMETER:

1) 1 digit numeric value (0 - 9).

2) 16 digit string (hex value).

PURPOSE:

This command allows the user to set a 1-to 8-byte repeating data pattern to be used during write/read testing. The length of the repeat is specified in the [x] argument while the actual data pattern is specified in the [yyyyyyyyyyyyyyyy] argument.

13. SET AUTO ERROR CHECK (1=enab) = [x]

PARAMETER:

1) 1 digit Boolean value. **Default = 1** (enabled).

PURPOSE:

This command is issued to enable or disable the clearing of CHECK CONDITION states. If a CHECK CONDITION occurs and the SET AUTO ERROR CHECK is enabled, a REQUEST SENSE will be issued to acquire the error codes from the device. Otherwise, no REQUEST SENSE is issued.

14. SET IDENT MSG ident=[xx] qbytes=[yyyy] (not supported in Orion)

PARAMETER:

1) 2 digit hexadecimal value (0 - FF).

2) 4 digit hexadecimal value (0 - FFFF).

PURPOSE:

This command sets the identify message [xx] for Tag Command Queing (TCQ), and the Que type and Que tag [yyyy]. These arguments can also be manipulated through the use of Variable Q0, with Q1 being the Identify message (C0h), Q2 being the Que type, and Q3 being the Que tag.

15. SET INITIATOR ID = [xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 15**.

PURPOSE:

Used to set the initiator ID in a multi-initiator system.

16. SET LUN = [xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 0**.

PURPOSE:

This command is used to set the Logical Unit Number (LUN) of the device being tested. The LUN number is inserted into all CDB's and Identify messages.

17. SET SCSI mode=[xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 0**.

PURPOSE:

This command controls the SCSI data transfer mode. When DISCONNECT is enabled, the device will assert it's ATTENTION signal at the end of the selection phase. The presumption is that the target device will respond with a message out phase. When the message out phase occurs, the port will respond with a IDENTIFY message which will have bit 6 set to a 1 (enable disconnect). The IDENTIFY command will include the LUN in the 3 least significant bits. The action of this command is also supported in synchronous modes of operation allowing synchronous transfers with and without disconnect.

The mode argument values are as follows:

00 = Narrow SCSI, disconnect disabled.

01 = Narrow SCSI, disconnect enabled.

02 = Wide SCSI, disconnect disabled.

03 = Wide SCSI, disconnect enabled.

17 = PPR negotiation for DT clocking, Wide SCSI, disconnect enabled.

97 = Packetized (IU), PPR negotiation for DT clocking, Wide SCSI, disconnect enabled.

18. SYNC OP MODE offset=[xxx] per=[yyy]

PARAMETER:

1) 3 digit decimal value (0 - 255) for Offset. **Default = 0**.

2) 3 digit decimal value (0 - 255) for Period. **Default = 0**.

PURPOSE:

This command negotiates with the SCSI device to determine the synchronous data transfer operating parameters. The first argument is the desired synchronous OFFSET value. A value of 0 implies (and selects) asynchronous operation. The maximum allowed offset value is 63.

The second argument is the minimum transfer PERIOD to be used between successive REQ pulses (or ACK pulses). The unit multiplier is 4 nanoseconds.

EXAMPLE:

3 Mbytes / Sec Offset = 6 Period = 83

4 Mbytes / Sec Offset = 6 Period = 62

4.5 Mbytes / Sec Offset = 6 Period = 55

5 Mbytes / Sec Offset = 6 Period = 50

10 Mbytes / Sec Offset = 6 Period = 25

20 Mbytes / Sec Offset = 6 Period =12

40 Mbytes / Sec Offset = 6 Period = 10

80 Mbytes / Sec Offset = 31 Period = 10 (25 ns) or 11 (33.3 ns)

160 Mbytes / Sec Offset = 62 Period = 9

320 Mbytes / Sec Offset = 62 Period = 8

Note: SCSI transfer rates are negotiated between devices with the maximum transfer rate allowed being the rate of the slowest device. The period value defines the maximum transfer rate for which the test port will negotiate. The target device determines the actual transfer rate.

19. SET #LBA TO XFER = [xxxx]

PARAMETER:

1) 4 digit decimal value (0 - 9999).

PURPOSE:

The transfer length set is also the variable X0. The value of TRANSFER LENGTH is used in all track based operations and controls the number of blocks transferred on each command issued to the SCSI device in the following tests:

READ CURRENT TRACK

WRITE CURRENT TRACK

WRITE INCR FORWARD

READ INCR FORWARD

WRITE RANDOM TRACK

READ RANDOM TRACK

It is also used as the increment value for the *INCREMENT HEAD* and *DECREMENT HEAD* commands.

When the TRANSFER LENGTH is set with this command, the test port limits the number of LBA's transferred to the size of the track buffer. Transfer length can also be set directly by setting the 'X0' variable. If not set, the transfer length is derived from the DUT via the CONFIGURE PORT command.

WARNING: When setting the 'X0' variable directly, the value is not checked and this method may result in a seek to an illegal LBA. Also, SECTORS PER CYLINDER is recalculated following this command.

20. SET TARGET ID = [xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 0**.

PURPOSE:

This command is issued to set the target ID number of the SCSI device being tested. A value of 16 causes the test port to search all available ID's from 0 to 15 during the *TEST SPIN UP* macro. If no SCSI devices are detected, a SELECTION TIMEOUT error will occur.

21. START QUE

PARAMETER:

8-character file name string

PURPOSE:

The *START QUE* (and *END QUE*) command is not currently implemented in Orion.

22. TEST PARITY

PARAMETER:

none

PURPOSE:

This command is a macro function that detects if the SCSI device properly handles PARITY during selection and arbitration phase. If the device detects parity (by not selecting) then there is no further action (test passed). If the device performs a selection regardless of invalid parity, a fatal PARITY ERROR is generated. If PARITY CONTROL = 0, then this test will always pass (parity generation / checking disabled).

23. TEST SYNC DATA XFERS

PARAMETER:

none

PURPOSE:

This command issues a SCSI READ EXTENDED command to the device. When the command enters the data transfer phase, ACK pulses are held off long enough to insure that REQ pulses are not being issued. At this time, the number of REQ pulses issued by the device is observed and printed on the test report as OBSERVED OFFSET.

6.2.1.9 CCS Command Sub-Group

40 Change Definition	1C Receive Diagnostics
39 Compare	14 Recover Buffer Data
18 Copy	17 Release Unit
3A Copy and Verify	03 Request Sense
Edit Vendor Unique Command	16 Reserve Unit
19 Erase	01 Rezero Unit
04 Format Unit	36 Lock/Unlock Cache
12 Inquiry (6)	31 Search Equal
1B Load/Unload	30 Search High
2B Locate (streaming cmd)	32 Search Low
4C Log Select	0B Seek (6)

40 Log Sense

15 Mode Select (6)

55 Mode Select (10)

1A Mode Sense (6)

5A Mode Sense (10)

08 Read (6)

05 Read Block Limits

3C Read Buffer

25 Read Capacity

37 Read Defect List

28 Read Extended (10)

3E Read Long

0F Read Reverse

13 Read Verify

07 Reassign Blocks

2B Seek Extended (10)

1D Send Diagnostics

33 Set Limits

11 Space

1B Start/Stop Unit

35 Synchronize Cache

00 Test Unit Ready

0A Write (6)

2E Write and Verify

3B Write Buffer

2A Write Extended (10)

10 Write Filemarks

3F Write Long

41 Write Same

The above is a list of SCSI Common Command Set commands. These commands function as templates where the op-code is entered for you, but you may add an additional 16 fields. The fields may be actual hex bytes, or you may wish to use some of the advanced features described in this Chapter such as variable substitution. Commas are used as field delimiters. A terminator finishes the command string. If you type a CR after entering a field, the editor inserts a simple terminator automatically. For example, to download a file with the command you would type a "Z" and this would serve as the terminator. In this example, you would have issued a **"SET FILE NAME"** command prior to issuing the CCS command. There are six variables: L, P, S, X, R, and N. Each variable is a double word data type, accessible by bytes (1-4). Three additional variables are available: ST,SK, and AD. In order to insert a variable into a command string, type the variable label in that field.

CCS commands can be issued to:

SCSI Direct access devices such as hard disk drives.

SCSI Sequential (streaming) access devices such as tape drives.

SCSI Optical memory devices.

SCSI WORM (Write Once/Read Multiple) devices.

SCSI Read Only/Direct Access devices.

SCSI Printers

SCSI Processors.

SCSI Scanner devices.

When the CCS OR VENDOR UNIQUE COMMAND field is selected, thirty-eight Common Command Set commands will appear on the host PC screen. These commands, such as TEST UNIT READY, SEEK, FORMAT, WRITE, etc. are frequently used to test SCSI disk drives.

⇨ NOTE: On any of the CCS commands that reference LBA, the user should use the L1-L4 variable. This is the variable that contains the current LBA.

⇨ NOTE: It is very important that you issue the appropriate command when testing different types of SCSI devices. For example, the "READ BLOCKS LIMITS" command is unique to SCSI sequential access devices such as tape drives, and should not be issued to other kinds of SCSI devices.

⇨ You will need a product manual for the SCSI device(s) being tested. This is because such commands as FORMAT, MODE SENSE, MODE SELECT, etc., have vendor unique information. Without such manuals, it will be difficult to create SCSI test programs.

The following is an example of a SEEK command listed under CCS OR VENDOR UNIQUE COMMAND:

COMMAND: SEEK ,0B,[XXXXXXXXXXXXXXXXXX]

COMMAND: SEEK ,0B,0,0,0,0,0

Byte 0 (0B) = the command's operating code.

Byte 1, 2, and 3 = the Logical Block Address in order from MSB to LSB.

Byte 4 = reserved.

Byte 5 = vendor unique.

⇨ NOTE: Commas must be used as field delimiters, as shown in the previous example.

The seven valid command terminators are Q, T, U, Z, Y, W, AND <<CR>> (see section 6.2.1.10). Any fields entered after these terminators will be ignored. "U" causes a file to be created from the data returned by the last command and uploaded to the server. The file name must be set prior to this operation. "Z" downloads and appends it to the command, again using the file name previously set. "Y" appends the current defect list and a header at the port. "T" sends the data to the system printer. "Q" appends the returned data to the current file on the server from the client.

If you wish to issue a command that is not listed in the SCSI command menu, perform the following:

1. Select the "EDIT CCS OR UNIQUE COMMAND", listed under SCSI COMMANDS, with the cursor key. Press the ENTER key.

2. "EDIT VENDOR-UNIQUE COMMAND" will appear at the top of the command listing. Press the ENTER key to select it.

3. After the words "VEN_UNQ," appear, enter the command's operation code (in hex) and the remaining bytes in the CDB field in either hex or with a labeled variable.

⇨ NOTE: If the operation code is a CCS command, the words "VENDOR-UNQ" will change to the command name.

A brief description of the SCSI commands listed under CCS OR VENDOR UNIQUE COMMANDS are listed on the following pages.

Change Definition (40h)

This command applies to all SCSI devices. It is issued to modify the operating definition of the device under test with respect to commands from the FlexStar tester.

Compare (39h)

This command applies to all SCSI devices. It is issued to compare data from one logical unit on the SCSI bus with another, or with the same logical unit. Bytes 3 (MSB) through 5 (LSB) are used to specify the byte length of parameters the device will send during the command's DATA OUT phase. If set to 0, no data will be transferred.

Copy (18h)

This command applies to all SCSI devices. It is issued to copy data from one SCSI device under test (source) to another (destination), for example, from a direct access to a sequential access device.

Copy and Verify (3Ah)

This command applies to all SCSI devices. It is similar to the "COPY" command, except the data written to the SCSI device under test can be verified after the WRITE operation. If the Byte Check bit 1 is set to 0, only medium verification (CRC, ECC, etc.) will be performed. If the Byte Check bit is set to 1, a byte-by-byte comparison of the written data will be performed

Edit Vendor Unique Command

This command applies to all SCSI devices. It allows the user to create and edit standard CCS commands or other vendor unique commands. Special terminators may be appended to the end of the command line (see Chapter 22 for a description of command terminators).

Erase (19h)

This command applies to Sequential Access SCSI devices. It is issued to erase all or part of the remaining medium from a current logical position of the sequential access SCSI device under test.

Format Unit (04h)

This command applies to Direct Access and Printer SCSI devices. It is issued to format the DUT using any one of five different formats. These formats correspond to the DEFECT LIST FORMAT field in the CDB's Byte 1, Bits 0-2.

IMPORTANT: Be sure to issue a "COMMAND COMPLETE TIMEOUT" statement before issuing a "FORMAT UNIT" command in your test program. This will give the UUT enough time to be completely formatted. The timeout period you set will depend on the storage capacity of the UUT.

Inquiry (6 byte) (12h)

This command applies to all SCSI devices. It requests that the device under test and any attached peripheral devices send the FlexStar SCSI test port information such as:

DEVICE TYPE

VENDOR IDENTIFICATION

PRODUCT IDENTIFICATION

PRODUCT REVISION LEVEL

VENDOR UNIQUE PARAMETERS

⇨ NOTE: If you issue a "CONFIGURE PORT" command in your test program, an "INQUIRY" command is automatically issued.

Lock/Unlock Cache (36h)

This command applies to Direct Memory Devices. It requests that the device under test disallow or allow logical blocks within a specified range to be removed from the cache memory by the device's cache replacement algorithm.

Log Select (4Ch)

This command applies to all device types. It provides a means for the tester to manage statistical information maintained by the device about the device or its logical units. Devices that implement the LOG SELECT command shall also implement the LOG SENSE command. The LOG SELECT command provides for sending log pages during a DATA OUT phase.

Log Sense (4Dh)

This command applies to all device types. It provides a means for the tester to receive statistical information maintained by the device about the device or its logical units. It is the complementary command to the LOG SELECT command.

Mode Select (6 byte) (15h)

This command applies to Direct Access, Sequential Access, Printer, WORM, Read Only (Direct Access), Scanner, and Optical Memory SCSI devices. It is issued by the FlexStar SCSI test port to specify the medium, logical unit, and/or peripheral device parameters to the DUT.

Mode Select (10 byte) (55h)

This command applies to Direct Access, Sequential Access, Printer, WORM, Read Only (Direct Access), Scanner, and Optical Memory SCSI devices. Same as MODE SELECT (6) except allows a larger parameter list.

Mode Sense (6 byte) (1Ah)

This command applies to Direct Access, Sequential Access, printer, WORM, Read Only (Direct Access), and Optical Memory SCSI devices. It is issued to the SCSI device under test to report its medium, logical unit, and peripheral device parameters to the FlexStar SCSI test port.

Mode Sense (10 byte) (5Ah)

This command applies to Direct Access, Sequential Access, printer, WORM, Read Only (Direct Access), and Optical Memory SCSI devices. Same as MODE SENSE (6) except allows a larger parameter list.

Read (6 byte) (08h)

This command applies to Direct Access, Sequential Access, WORM, Read Only (Direct Access), and Optical Memory SCSI devices. It is issued to transfer "x" number of adjacent logical blocks of data to the FlexStar SCSI test port. The amount of data transferred depends on the value entered in the TRANSFER LENGTH field located at Byte 4 in the CDB. If set equal to 0, 256 logical blocks will be transferred during the READ operation.

⇨ NOTE: If a READ command is issued in a SCSI test program and hardware data compare is performed, a WRITE command should be issued before the READ is executed. In addition, the logical block address should be the same in both commands.

Read Block Limits (05h)

This command applies to Sequential Access devices. It is issued by the test port to read the maximum and minimum block lengths (MSB to LSB) of the SCSI UUT.

Read Buffer (3Ch)

This command applies to all SCSI devices. It complements the "WRITE BUFFER" command and is issued to test the buffer memory of the device under test and the integrity of the SCSI bus. It is recommended that you issue a "RESERVE UNIT" command to the SCSI device under test before you issue the "READ BUFFER" command and a "RELEASE UNIT" command after the "READ BUFFER" command has been executed.

Read Capacity (25h)

This command applies to Direct Access, WORM, and Read Only (Direct Access) SCSI devices. The FlexStar SCSI test port issues this command to request information about the capacity of the DUT.

⇨ NOTE: The "GET WINDOW PARAMETERS" SCSI command's operating code is also 25 hex, and its CDB is 10 bytes long. This command is issued to SCSI Scanner devices.

Read Defect List (37h)

This command applies to Direct Access SCSI devices. It is issued to transfer the SCSI drive under test's defect lists to the FlexStar SCSI test port. If you set the Manufacturer's Defect List (MDL) and the Grown Defect List (GDL) bits to 1, the drive will transfer both lists to the test port. If both bits are set to 0, the drive will return only the Defect List Header.

Read Extended (10 byte) (28h)

This command applies to Direct Access, WORM, Read Only (Direct Access), Scanner, and Optical Memory SCSI devices. It is the same as the "READ" command, except its Logical Block Addresses occupy 4 bytes (MSB to LSB) and its transfer length is 2 bytes (MSB and LSB).

Read Long (3Eh)

This command applies to Direct Access SCSI devices. It is issued to the drive under test to transfer data to the FlexStar SCSI test port during the DATA IN phase.

Read Reverse (0Fh)

This command applies to Sequential Access SCSI devices. It is the same as the READ command, except data is transferred in a reverse direction.

Read Verify (13h)

This command applies to Direct Access, WORM, and Optical Memory SCSI devices. It is issued to the SCSI device under test to verify data written on its media. If the byte check bit (Byte 1) is set to 1, the logical blocks are read from the disk and compared with data transferred by the FlexStar SCSI test port. If the byte check is set to 0, the data is read from the disk and the ECC is checked for validity.

Reassign Blocks (07h)

This command applies to Direct Access, WORM, and Optical Memory SCSI devices. It is issued by the FlexStar test port to reassign defective blocks of the SCSI UUT to any physical address on the device reserved for that purpose. The command can be executed in one of two modes that must be specified in a MODE SELECT command, using the parameters located at Bytes 2 (MSB) through 3 (LSB). One mode is track oriented, and the other is cylinder oriented. If you select the track oriented mode (default mode), set the field equal to 1. If you select the cylinder-oriented mode, set it equal to the number of data heads in the SCSI device.

Receive Diagnostics (1Ch)

This command applies to all SCSI devices. It complements the "SEND DIAGNOSTICS" command, which must precede the "RECEIVE DIAGNOSTICS" command; otherwise, the SCSI device under test will report a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID COMMAND Sense key/Error Code. The command is issued to the device under test to report self-diagnostics data to the FlexStar SCSI test port.

Recover Buffer Data (14h)

This command applies to Sequential Access and Printer SCSI devices. It is issued to read data that has been transferred to the device under test's buffer, but has not been written to its medium.

Release Unit (17h)

This command applies to Direct Access, Sequential Access, Printer, WORM, Read Only (Direct Access), Scanner, and Optical memory SCSI devices. It is issued to release logical units previously reserved with the "RESERVE UNIT" command.

Request Sense (03h)

This command applies to all SCSI devices. It allows the FlexStar test port to request detailed information from the UUT, upon execution of a command. The test port supports both non-extended and extended sense data formats.

⇨ NOTE: If the test port receives a CHECK CONDITION code during test, it automatically issues a REQUEST SENSE command to the SCSI device under test.

Reserve Unit (16h)

This command applies to Direct Access, Sequential Access, printer, WORM, Read Only (Direct Access), Scanner, and Optical memory SCSI devices. It is issued to reserve logical units (devices under test) for exclusive use. When used with the "REVERSE UNIT" command, it provides a means of resolving contention on the SCSI bus in multiple initiator systems.

⇨ NOTE: The FlexStar SCSI test ports and the FlexStar host software also support the command's extent reservation option.

Rezero Unit (01h)

This command applies to Direct Access, WORM, Read Only (Direct Access) and Optical Memory SCSI devices. It recalibrates the DUT to Track 0.

Search Equal (31h)

This command applies to Direct Access, WORM, and Read Only (Direct Access) SCSI devices. It is the same as "SEARCH HIGH", except the data in the logical record is the same as data in the pattern, if the inverter bit is set to 0. If it is set to 1, the data in the logical record is not equal to the data in the pattern.

Search High (30h)

This command applies to Direct Access, Worm, and Read Only (Direct Access) SCSI devices. Issue this command to search one or more logical blocks of the SCSI device under test for equality or inequality to a data pattern. If the inverter bit (Byte 1, Bit 4) is set to 0, the search argument descriptor(s) shall be satisfied that the data in the logical record is greater than the data in the pattern. If set to 1, the data in the logical record is not equal to the data in the pattern.

Search Low (32h)

This command applies to Direct Access, WORM, and Read Only (Direct Access) SCSI devices. It is the same as "SEARCH HIGH", except the data in the logical record is less than the data in the pattern, if the inverter bit is set to 0. If it is set to 1, the data in the logical record is greater than or equal to the data in the pattern.

Seek (6 byte) (0Bh)

This command applies to Direct Access, WORM, Read Only (Direct Access), and Optical Memory SCSI devices. It is issued by the FlexStar SCSI test port to the DUT to seek to a specific logical block address located at the CDB's Bytes 1 (MSB) through 3 (LSB).

Seek Extended (10 byte) (2Bh)

This command applies to Direct Access, WORM, Read Only (Direct Access), and Optical Memory SCSI devices. It is similar to the "SEEK" command, except its Logical Block Addresses occupy 4 bytes (MSB to LSB).

Send Diagnostics (1Dh)

This command applies to all SCSI devices. It is issued to the device under test to perform diagnostics on itself, on any attached peripheral devices, or both.

Set Limits (33h)

This command applies to Direct Access and Optical Memory devices. It defines the range within which subsequent linked commands may operate.

⇨ Note: A second SET LIMITS command may not be linked to a chain of commands in which a SET LIMITS command has already been issued.

Start/Stop Unit (1Bh)

This command applies to Direct access, WORM, and Read Only (Direct Access). It requests that the DUT enable or disable the logical unit for further operation. A start bit of 1 requests the logical unit be made ready for use. A start bit of 0 requests the logical unit be stopped.

Synchronize Cache (35h)

This command applies to Direct Access devices. It ensures that logical blocks in the cache memory, within the specified range, have their most recent data value recorded on the physical medium. If a more recent data value for a logical block within the specified range exists in the cache memory than on the physical medium, then the logical block from the cache memory shall be written to the physical medium.

⇨ NOTE: Logical blocks are not necessarily removed from the cache memory as a result of the synchronize cache operation.

Test Unit Ready (00h)

This command applies to all SCSI devices. It verifies that the SCSI device under test is ready. This should be one of the first commands you issue when creating a SCSI test program.

⇨ NOTE: if you issue a “CONFIGURE PORT” command in your test program, the FlexStar test port to clear any power up CHECK CONDITION states automatically issues two “TEST UNIT READY” commands.

Write (6 byte) (0Ah)

This command applies to Direct Access, Sequential Access, WORM, and Optical Memory SCSI devices. It is issued to the device to write data transferred by the FlexStar SCSI test port. The amount of data written depends on the value entered in the TRANSFER LENGTH field located at Byte 4 in the CDB. If you set it equal to 0, 256 logical blocks will be transferred during the WRITE operation.

Write and Verify (2Eh)

This command applies to Direct Access, WORM, and Optical Memory SCSI devices. It is issued by the FlexStar SCSI test port to write data to the DUT's Logical Block Addresses (MSB to LSB), and then verify the data. If Byte 1's byte check is set to 1, the transferred data will be checked on a byte-to-byte basis.

Write Buffer (3Bh)

This command applies to all SCSI devices. It complements the "READ BUFFER" command. It is issued to test the buffer memory of the device under test and the integrity of the SCSI bus. If you wish to determine the maximum amount of data that can be transferred from the device's buffer memory, issue a "READ BUFFER" command and set the allocation length (Bytes 6-8) to 4.

⇨ NOTE: The maximum data transfer length is dependent on the specific model of the port card in use, the size of the FlexStar test port buffer.

Write Extended (10 byte) (2Ah)

This command applies to Direct Access, WORM, and Optical Memory SCSI devices. It is the same as "WRITE" command, except its Logical Block Addresses occupy 4 bytes (MSB to LSB) and its transfer length is 2 bytes (MSB and LSB).

Write Long (3Fh)

This command applies to Direct Access SCSI devices. Issue this command to the drive under test to write data transferred by the FlexStar SCSI test port to its medium. After issuing this command, issue a "REQUEST SENSE" command. The correct amount of bytes can be determined from the information returned in the command's extended sense data bytes.

Write Same (41h)

This command applies to Direct Access SCSI devices. It requests that the device under test write the single block of data to the medium multiple times.

6.2.1.10 Command Terminators

COMMAND TERMINATORS are used with VENDOR UNIQUE or BUILD CDB command strings to define the behavior of the data transfer mode. The following terminators are available:

I - Transfers data to the TRACK BUFFER from the device.

O - Transfers data from the TRACK BUFFER to the device.

Q - Appends data from the device to a named file on the SERVER. See *FILENAME* command.

T - Uploads data from the device directly to the system printer attached to the SERVER.

U - Uploads data from the device to a named file on the SERVER (overwrites any existing file) See *FILENAME* command.

W - Transfers data from a previously selected buffer to the device.

Y - Transfers the client DEFECT list to the device.

Z - Downloads data from the current filename to the device See *FILENAME* command.

EXAMPLES FOR SCSI:

```
FILENAME = CACHEOFF.RAM

VEN_UNQ MODE SL,15,00,00,00,14,00,Z

VEN_UNQ MODE SN,1A,00,37,00,FF,00,T
```

Mode select command downloads the file “CACHEOFF.RAM” created by using the binary file editor. Mode sense reads page code 37 and sends it to the printer. The user can then check the cache bit.

6.2.1.11 Set Parameter Commands

Set Cmd Complete Timeout

Set Data Pattern (2 byte)

Set Data Pattern (12 byte)

Set Format Type

Set MIN Cylinder

Set MAX Cylinder

Set MIN Head

Set MAX Head

Set Model Number

Set Random Data Pattern

Set Retries

Set Serial Number

Set Write Verify Mode

1. SET CMD TIMEOUT time(sec) = [xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). Default = 10.

PURPOSE:

This command is issued to set the maximum amount of time delay before the test ports will timeout following a command phase and no SCSI bus activity is detected.

This command is useful in setting timeout values for commands that are expected to take a long time to complete. An example is the SCSI FORMAT UNIT command.

2. SET DATA PATTERN-2 byte(x) = [xxxx]

PARAMETER:

1) 4 digit hexadecimal value (0 - FFFF). **Default = 4E65(hex)**.

PURPOSE:

This command is used to initialize the TRACK BUFFER with a fixed 2-byte repetitive data pattern for use in writing or read data compare with a previously written pattern.

3. SET DATA PAT-12Byt(x)=[xxxxxxxxxxxxxxxxxxxxxxxx]

PARAMETER:

1) 24 digit hexadecimal value.

PURPOSE:

This command is used to initialize the TRACK BUFFER with a fixed 12-byte repetitive data pattern for use in writing or read data compare with a previously written pattern.

4. SET FORMAT TYPE CODE = [xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 0**.

PURPOSE:

This command controls the read and write modes of the Orion test ports via the FORMAT TYPE CODE:

IDE TEST PORTS

TYPE 0 - Default mode, all read and write commands are normal.

TYPE 1 - The test port will issue a READ LONG or a WRITE LONG for each read or write issued.

TYPE 40 - Used to assign bad blocks. NOTE: Not supported by all drives; see manufacturers specifications.

TYPE 99 - The test port will issue FORMAT TRACK commands to the device (IDE only) with the appropriate formatting data as specified in the CAM specification. All blocks are marked as good. NOTE: A drive may accept the command but not do anything. See drive manufacturers specification.

Note: This mode only works with the WRITE INCR FORWARD command.

SCSI TEST PORTS

The following format codes are used in the READ DEFECT LIST macro and SCSI FORMAT UNIT command as follows:

TYPE 8 - GList in LBA format.

TYPE 13 - GList in CHS format.

TYPE 16 – PList in LBA format.

TYPE 21 - PList in CHS format.

TYPE 24 - Both Glist & PList in LBA format.

TYPE 29 - Both Glist & PList in CHS format.

5. SET MIN CYLINDER cyl=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). **Default = 0**.

PURPOSE:

This command sets the MINIMUM operating range for the cylinders of the device under test. The default is 0 but the minimum value may be set to any number less than MAX CYLINDER. This value is reset to zero when the *CONFIGURE PORT* macro is executed.

Associated variables: MN.

6. SET MAX CYLINDER cyl=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535). **Default = 0**.

PURPOSE:

This command sets the MAXIMUM operating range for the cylinders of the device under test. The default MAXIMUM CYLINDER is usually set by the *CONFIGURE PORT* macro but can be overwritten by this command.. The maximum value can be any value greater than the MIN CYLINDER value.

Associated variables: MX.

7. SET MIN HEAD head=[xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 0**.

PURPOSE:

This command sets the MINIMUM HEAD number when operating the device in physical mode. At the beginning of the test, or following a *CONFIGURE PORT* command, the default value is set to 0. This command can be used to override the initial value and can be any value that is less than MAX HEAD.

Associated variables: HN

8. SET MAX HEAD head=[xx]

PARAMETER:

1) 2 digit decimal value (0 - 99). **Default = 3**.

PURPOSE:

This command sets the MAXIMUM HEAD number when operating the device in physical mode. The MAX HEAD value is initially set by the *CONFIGURE PORT* command. This command can be used to override the initial value and can be any value that is greater than MIN HEAD.

Associated variables: HX

9. SET MODEL num=[xxxxxxxxxx]

PARAMETER:

1) 10 digit ASCII string. **Default = NULL**.

PURPOSE:

The SET MODEL NUMBER command allows the user to enter a 10-digit ASCII string that will override the model number string acquired by the *CONFIGURE PORT* command. The new model set here will appear on the test report.

Variables affected: MM

10. SET RANDOM DATA PATTERN

PARAMETER:

none

PURPOSE:

This command sets the random data mode where a psuedo-random pattern is used to fill the TRACK BUFFER prior to writing or reading. Since the pattern is a psuedo-random pattern, the same sequence of random numbers is generated each time this command is invoked. This command is the same as *SET DATA MODE* = 1 which fills each sector with a series of random numbers based on an initial fixed seed.

11. SET RETRIES wrt=[xxx] read=[yyy]

PARAMETER:

1) 3 digit decimal value (0 - 255). **Default = 2**.

2) 3 digit decimal value (0 - 255). **Default = 2**.

PURPOSE:

This command is used to set the retry parameters of the test port for read retries and write retries. After an error is detected, READ retries are utilized first followed by write retries. If the error is detected a number of times that are equal to the CREDENCE value, retries are stopped and the next command will be issued.

12. SET S/N=[xxxxxxxxxxxxxxxx]

PARAMETER:

1) 16 digit ASCII string.

PURPOSE:

The SET SERIAL NUMBER command allows the user to enter a 16 digit ASCII string that will override the serial number string acquired by the *CONFIGURE PORT* command. The new serial set here will appear on the test report and is used to name the REPORT file and DAT file.

Variables affected: SN

13. SET WRITE VERIFY (0=off, 1=on) = [x]

PARAMETER:

1) 1 digit Boolean value. **Default = 0** (disabled).

PURPOSE:

This command, if enabled, causes all write operations to perform a read-after-write which verifies the data immediately after it has been written. The reads will be of the same length as the writes and the read data will be compared to the write data if DATA COMPARE is enabled.

6.2.1.12 Seek Commands

Decrement Cylinder

Decrement Head

Increment Cylinder

Increment Head

Seek HOME (recalibrate)

Seek to Cylinder

Seek Incr Forward

Seek Incr Reverse

Seek Random

Seek Oscillate

Seek Alt Full Stroke

Select Head

Set Seek Increment

1. DECREMENT CYLINDER

PARAMETER:

none

PURPOSE:

This command decrements the current cylinder value by the amount set in the SET SEEK INCREMENT value. If the device is positioned at the MIN CYLINDER, a seek to the MAX CYLINDER will be performed. A *SEEK TO CYLINDER* command is then issued to place the heads over the new cylinder.

2. DECREMENT HEAD

PARAMETER:

none

PURPOSE:

This command decrements the current head address by one. When MIN HEAD has been reached, the head address rotates to MAX HEAD. A *SEEK TO CYLINDER* command is issued to change the head address but the cylinder address is not changed.

3. INCREMENT CYLINDER

PARAMETER:

none

PURPOSE:

This command increments the current cylinder value by the amount set by the SET SEEK INCREMENT value. If the device is positioned at the MAX CYLINDER, the next seek will be to the MIN CYLINDER value. A *SEEK TO CYLINDER* command is then issued to place the heads over the new cylinder.

4. INCREMENT HEAD

PARAMETER:

none

PURPOSE:

This command increments the current head address by one. When MAX HEAD has been reached, the head address rotates to MIN HEAD. A *SEEK TO CYLINDER* command is issued to change the head address but the cylinder address is not changed.

5. SEEK HOME (RECAL)

PARAMETER:

none

PURPOSE:

This command causes the test port to issue a SEEK HOME (RECALIBRATE) command to the device. The interface specific command will cause the device heads to be positioned at CYLINDER 0. Note that many device types do not actually seek even though the command is accepted.

6. SEEK TO CYLINDER cyl=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command issues a SEEK command to the device to move the heads to the specified CYLINDER. Note that many devices do not actually perform a seek even though the command is accepted.

7. SEEK INCR FORWARD pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command causes the device to SEEK incrementally from MIN CYL to MAX CYL using the increment value set by the SET SEEK INCREMENT command. The ***pass*** argument specifies the number of times this function will be performed.

8. SEEK INCR REVERSE pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command causes the device to SEEK reverse from MAX CYL to MIN CYL using the increment value set by the SET SEEK INCREMENT command. The **pass** argument specifies the number of times this function will be performed.

9. SEEK RANDOM CYLINDER pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command causes a random SEEK to be performed on the device. The random number is generated by a psuedo-random number generator and is range limited to be between MIN CYL and MAX CYL.

10.SEEK OSCILLATE pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command causes an OSCILLATING SEEK to be performed on the device. The software begins by seeking to the middle cylinder (MAX CYL – MIN CYL) and then begins to seek to cylinder MID-1, MID+1, MID-2, MID+2, and etc. until the MIN and MAX cylinders have been reached.

11.SEEK ALT FULL STROKE pass=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command will cause the device to SEEK between MIN CYL and MAX CYL continuously until the ***pass*** count has decremented to zero.

12.SELECT HEAD hd=[xx]

PARAMETER:

1) 2 digit decimal value (0 - 99).

PURPOSE:

This command allows an individual physical head to be selected with the specified head. The number of seeks counter will not be incremented as actual seeks are not performed.

13.SET SEEK INCREMENT inc=[xxx]

PARAMETER:

1) 3 digit decimal value (0 - 255). **Default = 1**.

PURPOSE:

This commands argument is used to specify the number of cylinders to move each time an increment or decrement SEEK is performed.

6.2.1.13 Test Macro Commands

} End Performance

Profile Current

Start Performance {

Test Average Seek Timing

Test Drive Spin Up Time

Test Index Timing

Test OSC Avg Seek Timing

Test IDE I/O

Test Seek Timing

Test UDMA

1. } END PERFORMANCE

PARAMETER:

none

PURPOSE:

This command is used to define the END of a START PERFORMANCE block. When this command is encountered, data gathering is halted and the results are appended to a .PRF file.

2. PROFILE CURRENT event=[xxxx] rate(us)=[yyyyy]

PARAMETER:

1) Up to 4 ASCII characters.

2) 5 digit decimal value (0 - 65535).

PURPOSE:

This command allows the devices power to be sampled and stored in a disk file buffer on the SERVER. The first argument in the command is the EVENT string, which defines the trigger that starts sampling. Valid EVENT mnemonics are as follows:

I - Take sample Immediately (asynchronous).

O - Take sample when power is being turned OFF.

P - Take sample when power is being turned ON.

Cxx - Take sample when [C]ommand xx is issued.

If the EVENT string ends with the character 'W', the results will be written to a database file called RxxPxxDx.PWR where Rxx = Rack#, Pxx = Port#, and Dx = Drive#. Using this convention, a power profile database is created for each test port.

Examples:

PROFILE CURRENT event=IW rate (us)=500

Takes an immediate sample (when the command is executed) at a 500-usec rate and creates a database file.

PROFILE CURRENT event=C30W rate (us)=1000

Takes a sample beginning when the command 30(hex) is issued, and at a 1000 usec rate and creates a database file.

The second argument defines the sample rate in microsecond increments. The actual value is rounded to the nearest millisecond interval with 1 millisecond being the minimum sample rate allowed. For example, a sample rate of 900 microseconds will actually yield a 1 millisecond sample rate. The older 2450 and PC104 power cards have a 4 millisecond minimum sample rate.

Sampling begins when the specified EVENT is detected. Sampling will continue until the sample buffer is full (2048 samples of each channel).

3. START PERFORMANCE { [xxxxxxxxxxxxxxxxxxxx]

PARAMETER:

1) 20 digit ASCII string.

PURPOSE:

This command defines the beginning of a performance measurement block. Data gathering begins with this command and ends with the *END PERFORMANCE* command. All data transfers within the START / END PERFORMANCE block are analyzed for throughput times and will appear on the test report.

The ASCII string argument is used as a header on the report file to identify the purpose of the performance measurement.

4. TEST AVERAGE SEEK TIMING #seeks=[xxxxx]

PARAMETER:

1) 5 digit decimal value (0 - 65535).

PURPOSE:

This command is a complete macro that will perform a random seek test and acquire the MIN, MAX, and AVERAGE values for the specified number of seeks performed [xxxxx].

5. TEST DRIVE SPIN UP TIME

PARAMETER:

none

PURPOSE:

This command is a macro that applies power to the device and measures time to ready and current usage during start up. The program flow for interface specific devices is as follows:

IDE INTERFACE

Clear measurement variables.

Enable power to the device.

Wait for variable “SD” Delay

Wait for drive BUSY to be de-asserted

Write to SEL/HEAD register to select device

Write device CONTROL register to issue a soft reset

Wait for drive BUSY to be de-asserted

Issue a command to seek to CYL 0, SECTOR 1

Wait for drive ready status (50 hex).

SCSI / ATAPI INTERFACE

Clear measurement variables.

Enable power to the device.

ARBITRATION and SELECTION phase.

Issue START command.

Issue TEST UNIT READY command and wait for drive ready.

- Test Spinup Power Sampling
    - Default sample rate for power profiling is 8 ms. This can be modified using the “PROFILE CURRENT” command.
    - Power is turned ON to the DUT and triggers the power profile background task to wakeup every sample interval and sample the current of the 5V and 12V channels. The results are stored into a sample buffer.
    - Meanwhile, the foreground task is waiting for the DUT to become ready.
    - Complete the sampling of a maximum of 2048 samples.
    - Calculate MIN, MAX, and AVG values from the sample buffer.
    - Challenge MIN, MAX, and AVG against the abort limits if enabled.

6. TEST INDEX TIMING (not supported in Orion)

PARAMETER:

none

PURPOSE:

This macro measures and reports the rotational spindle speed of the device. Test results will appear on the output report under RPM. This test only functions on IDE devices that support the INDEX bit in the status register.

7. TEST OSC SEEK TIME

PARAMETER:

none

PURPOSE:

This macro command will perform an OSCILLATING SEEK and measure the MIN, MAX, and AVERAGE of all seeks performed. The accumulated values are provided on the report output. The actual number of seeks performed depends on the drive geometry.

8. TEST IDE I/O test=[x]

PARAMETER:

1) 1 digit decimal value (0 - 9).

PURPOSE:

This command will perform various IDE only macro tests. The ***test*** argument provides the test macro function as follows:

0 - Test IOIS16 DEASSERTED (8 bit mode). This test is performed as follows:

Read the STATUS register and test if IOIS16 is de-asserted during the 8-bit transfer. Test will fail for a “IOIS16 ASSERTED DURING 8 BIT XFER” error if IOIS16 is asserted during a 8 bit data transfer.

1 - Test IOIS16 ASSERTED (16 bit mode). This test is performed as follows:

Issue an IDENTIFY DRIVE command and test if IOIS16 is asserted during the data phase of this command. Test will fail for a “NO IO16” error if IOIS16 is not asserted during a 16 bit data transfer.

9. TEST SEEK TIMING

PARAMETER:

none

PURPOSE:

This command is a complete test macro that performs three discrete seek timing tests and provides the results to the test report:

1 track seek timing. Seeks from MIN CYL to MIN CYL +1.

1/3 full stoke seek timing. Seeks from MIN CYL to MAX CYL / 3.

Full stroke seek timing. Seeks from MIN CYL to MAX CYL.

Each of these tests are done 100 times and the average taken for each. The results are then provided on the test report.

10. TEST UDMA

PARAMETER:

none

PURPOSE:

This command is included for compatibility with the legacy systems but is not supported in Orion.

6.2.1.14 Voltage Margin Commands

Drive Power ON

Drive Power OFF

Set Ch0 Volts (5V channel)

Set Ch1 Volts (12V channel)

Set Power Slew Rate

1. DRIVE POWER ON

PARAMETER:

none

PURPOSE:

This command will enable power to the device, based on the values set by the SET CH0 VOLTS and the SET CH1 VOLTS commands. If no values have been set prior to this command, CH0 and CH1 will be set to nominal (5.00V and 12.00V). The SLEW RATE can also be controlled to provide variable ramp rates at power on. Power is automatically turned OFF at test end but can be optionally left on by enabling this feature in the Host setup menu.

2. DRIVE POWER OFF

PARAMETER:

none

PURPOSE:

This command removes power to the device immediately. The time that the voltage takes to decay to zero is load dependent. This is because the output stage of the power regulator will go to a high impedance state when the regulator is turned off.

Note: Always use a WAIT DELAY prior to turning off power when using a write command as there may still be data being written to the DUT.

3. SET CH0 VOLTS = [xxxxx] Reg# = [y]

PARAMETER:

1) 5 digit ASCII string. **Default = 5.00**.

2) 1 digit numeric value (0 - 9). **Default = 0**.

PURPOSE:

This command allows the user to set the CH0 (5V) voltage to the device. This command can be used to set a target voltage prior to enabling power or to modify the voltage output immediately if power is already enabled.

Power is enabled by using either the DRIVE POWER ON or TEST DRIVE SPINUP commands.

If power is OFF, this command will set up the initial target voltage level to be used when power is turned on. If power is already ON, this command will cause the voltage to change to the new value immediately.

The first argument is a 5 digit ASCII string used to define the voltage (in volts) and can include a decimal point. The second argument is a REG number used for addressing multiple power modules (not used in Orion test systems).

4. SET CH1 VOLTS = [xxxxx] Reg# = [y]

PARAMETER:

1) 5 digit ASCII string. **Default = 12.00**.

2) 1 digit numeric value (0 - 9). **Default = 0**.

PURPOSE:

This command allows the user to set the CH1 voltage to the device. This command can be used to set a target voltage prior to enabling power or to modify the voltage output immediately if power is already enabled.

Power is enabled by using either the DRIVE POWER ON or TEST DRIVE SPINUP commands.

If power is OFF, this command will set up the initial target voltage level to be used when power is turned on. If power is already ON, this command will cause the voltage to change to the new value immediately.

The first argument is a 5 digit ASCII string used to define the voltage (in volts) and can include a decimal point. The second argument is a REG number used for addressing multiple power modules (not used in Orion test systems).

5. SET POWER SLEW RATE (x1ms) Ch0=[xxx] Ch1=[yyy]

PARAMETER:

1) 3 digit decimal value (0 - 255) for Ch 0. **Default = 0**.

2) 3 digit decimal value (0 - 255) for Ch 1. **Default = 0**.

PURPOSE:

This command allows a controlled slew rate (time from zero to nominal voltage) for each of the two power channels. The slew rate is controlled in 1-millisecond increments. Slew rate is load dependent and therefore will be approximate.

6.3 VARIABLES

Variables are used in the Orion script language to perform temporary data storage and control functions. Some variables are general purpose, but most variables serve program specific functions. While most variables are 4 byte (32 bit) numeric types, some variables are string types. The following table describes the supported variables and their usage:

Description	*Variable*	*Interface*	*Size (bytes)*
General Purpose	A0,B0,C0,D0,F0,G0, H0,I0,J0,K0,M0,N0, O0,PO,U0,W0,WA, WB,WC,WD,WE,W F,Y0,Z0,ZA,ZB,ZC	ALL	4
Additional Sense Code Qualifier	AQ	SCSI, ATAPI	1
Additional Sense Code	AD	SCSI, ATAPI	1
Board Control	BC	ALL	1
Blocks Read High	BH	SCSI, ATAPI	4
Blocks Read Low	BL	SCSI, ATAPI	4
Butterfly Seek	BS	ALL	2
Vendor Unique Transfer Length. (If non-std)	BX	ALL	2
Enable Command Block Reg (ATA)	CB	IDE	1
CRC Count (ATA)	CC	IDE	4
Current CYL Value	CN	IDE	4
Number of CRC retries	CR	IDE	1
Command Time	CT	SCSI, ATAPI	4
CH0 voltage	CV	ALL	4
Capability Word from the ATAPI Identify Word 49	CW	ATAPI	2
Drive byte count	DA	IDE	2
Data Buffer	DB	ALL	4
Download Length	DL	ALL	4
Device ID	DI	ALL	16
DRQ type	DT	IDE	1
48 bit LBA (ms 16 bits)	E0	IDE	4
Device Error Codes	EA, EB, EC	ALL	4
Compare Error Location	EL	ALL	4
Legacy error type	ET	ALL	1
Error Handling Variables	FA-FH	SCSI	1
File Pointer	FP	ALL	4
Fibre channel speed variable	FS	SCSI	1
Simulated Error Variable	FU	ALL	4
Current Head	HD	IDE	2
Host Interval	HI	ALL	4

Min head number	HN	ALL	1
Max head number	HX	ALL	1
IOC Status	IS	SCSI	1
Current LBA	L0 (L1-L4)	ALL	4
LBA's Per Track	LT	SCSI	4
Loop Counters	LA,LB,LC	ALL	4
Log (Error List) Entries	LE	ALL	2
Link Status C0 Link Failure count (Parity Test Result, Fibre Channel Only)	LH	SCSI	4
Link Status C1 Loss of Sync count (Parity Test Result, Fibre Channel Only)	LI	SCSI	4
Link Status C2 loss of Signal count (Parity Test Result, Fibre Channel Only)	LJ	SCSI	4
Link Status C3 primitive sequence error (Parity Test Result, Fibre Channel Only)	LK	SCSI	4
Link Status C4 Invalid transmission word (Parity Test Result, Fibre Channel Only)	LL	SCSI	4
Link Status C5 Invalid CRC count (Parity Test Result, Fibre Channel Only)	LM	SCSI	4
Current script line number	LN	ALL	4
Timed loop value	LT	ALL	4
Logical LBA’s per cylinder	LX	ALL	4
Model number	MM	ALL	16
Minimum LBA	MN	SCSI, ATAPI	4
Min LBA Ext (48 bit mode)	MO	IDE	4
MACRO - Pass Counts	MP	ALL	4
Middle LBA	MR	IDE	4
"STEP MODE" variable	MS	ALL	1
Maximum LBA	MX	SCSI, ATAPI	4
Max LBA Ext (48 bit mode)	MY	IDE	4
Cylinder Value	N0	ALL	4
Network Address	NA	ALL	16
Number of blocks between seeks	NB	ALL	2
Number of commands	NC	ALL	4
Number of heads	NH	ALL	1

Number of cylinders	NL	ALL	2
SCSI SDTR Negotiated Value	NN	SCSI	4
Number of seeks	NS	ALL	4
Open & Close delay	OC	ATAPI	2
Resulted Option/Feature set after PPR negotiation (SCSI)	OF	SCSI	1
Seek Overhead	OH	ALL	4
Oven temperature	OT	ALL	2
Oven humidity	OU	ALL	2
Preceding Command (ATA)	PC	IDE	1
Port Model Number	PM	ALL	16
Port Number	PN	ALL	1
General Purpose	PO	ALL	4
Port Type	PT	ALL	4
NPM Version #	PV	ALL	4
Serial number string	QA – QE	ALL	4
Reminder of Division Operation	RM	ALL	4
Minimum Random transfer length	RN	ALL	4
Random Number	R0	ALL	4
UDMA CCTO retry command timeout time	RR	IDE,ATAPI	4
Random Number Seed	RS	ALL	4
Compare Result	RT	ALL	1
Maximum Random transfer length	RX	ALL	4
Sector Number	S0 (zero)	ALL	4
Seek Count	SC	ALL	2
Spin Up Delay	SD	ALL	2
Command time HIGH	SH	ALL	4
Seek increment	SI	ALL	4
Sense Key From Last Request Sense	SK	SCSI, ATAPI	1
Command time LOW	SL	ALL	4
Mode Save	SM	SCSI	4
Serial Number	SN (string)	ALL	16
Serial Number Offset	SO	ALL	2
Spinup Time Result	SP	ALL	4
Sector Size	SS	SCSI, ATAPI	4
Error Status	ST	ALL	4
LBA’s per cylinder	SX	ALL	2
LBA’s per track	SY	ALL	2
'TE' Sense Dump Threashold	TE	SCSI	1
Total Blocks Transferred High	TH	ALL	4

Total Blocks Transferred Low	TL	ALL	4
Target ID Number	TN	SCSI, ATAPI	2
Current Temperature Reading	TP	ALL	4
Total User Timeout	TT	ALL	2
Temperature – CPU (Pegasus system only)	TU	ALL	1
Temperature - Socket Z Pegasus system only	TZ	ALL	1
Failure Code	UC	ALL	4
User Timeout(usec)	UT	ALL	4
Elapsed Time	VA-VB	ALL	4
ATAPI Revision Field	VA, VB	ATAPI	4, 4
5V Channel Value (Mv)	VC	ALL	4
12V Channel Value (Mv)	VD	ALL	4
Vendor Option Bits	VO	ALL	4
General Variable	W0	ALL	4
General Variable	WA	ALL	4
General Variable	WB	ALL	4
General Variable	WC	ALL	4
General Variable	WD	ALL	4
General Variable	WE	ALL	4
General Variable	WF	ALL	4
Wide mode	WM	SCSI	1
Transfer Length	X0	ALL	4
General Variable	Y0	ALL	4
Alternate status register	YA	IDE,ATAPI	1
Drive address register	YB	IDE,ATAPI	1
Sector count register	YC	IDE,ATAPI	1
Drive/head register	YD	IDE,ATAPI	1
Error register	YE	IDE,ATAPI	1
Feature Register	YF	IDE,ATAPI	1
Cylinder high register	YH	IDE,ATAPI	1
Cylinder low register	YL	IDE,ATAPI	1
Sector number register	YN	IDE,ATAPI	1
Command Register	YO	IDE,ATAPI	1
Status register	YS	IDE,ATAPI	1
General Variable	Z0	ALL	4
General Variable	ZA	ALL	4
General Variable	ZB	ALL	4
General Variable	ZC	ALL	4

6.3.1 VARIABLE DETAILS

A0 (A1-A4),B0,C0,D0,F0,G0,H0,I0,J0,K0,M0,N0,O0,P0,U0,W0,Y0,Z0

General-purpose variable. This variable may be used as a single 32 bit variable or four 8-bit variables with A1 containing the LSB and A4 containing the MSB.

AD

Additional sense code from the last REQUEST SENSE command (SCSI).

AM

ATA Mode (ATA).

AQ

Additional sense code qualifier from the last REQUEST SENSE command (SCSI).

BC

Board control variable.

BH

Blocks Read High double word contains the Most Significant 32 bits of the blocks read counter.

BL

Blocks Read Low double word contains the Least Significant 32 bits of the blocks read counter.

BS

Butterfly seek.

BX

Sets the vendor unique transfer length. This overrides the transfer length implied in the command block.

CB

Enable command block register (ATA).

CC

Additional sense code from the last REQUEST SENSE command (SCSI).

CN

Current Cylinder value for IDE interface.

CR

Defines the number of times a CRC error is retried before being included in the error log.

CT

Command time variable. This variable accumulates time duration when executing seek timing or performance tests.

CV

Ch0 voltage in millivolts.

CW

Capability word (ATAPI) – (DMA flag bit only)

DA

Drives byte count.

DB

Debug flags

0x0002	// SCSI chip trace
0x0004	// Cause minidrivers to just return
0x0008	// WR/RD buffer combine mode
0x0010	// Get Temperature reading from TP instead of from Oven
0x0020	// Transfer ECC for Read/Write Long in 8 bit instead of 16
0x0040	// Enable Data Compare for Performance Test
0x0080	// Disable Data Generation when Data Compare is disabled
0x0100	// Do not sort defect list
0x0200	// Fall through the Goto if Timed Loop expired.
0x0400	// Continue operation even specified condition(s) has not met

DI

Device ID (interface).

DL

Download Length gets or sets the length of a file download. When a file is uploaded from the client, the DL variable contains the length of the transfer.

EXAMPLE:

<008> SET FILENAME =READSEC1.DAT

<009> VEN UNQ 28,00,00,00,00,00,00,01,00,U

<010> WAIT DELAY (x.1 sec)=30

<011> SET VARIABLE v=DL (x)n=00000200 ; force download file size

<012> SET FILENAME =READSEC2.DAT

<013> UPLOAD FILE

DT

DRQ type.

E0

Used to extend the L0 variable for 48 bit IDE LBA mode.

EA, EB, EC

Device error codes. The following table shows the possible values of these variables following an error condition. Note also that these error codes are legacy compatible.

Medium Related Drive Errors			
Variable:	**EA**	**EB**	**EC**
Read Error	Obsolete		
RNF Error	00	00	01
Seek Error	00	00	02

Write Fault	00	00	03
Head Select Error	Obsolete		
Primary Error	00	00	06
Scratch Error	Obsolete		
Grown Error	00	00	08
ID CRC Error	00	00	09
No Address Mark Found	00	00	0C
Data CRC Error	00	00	0D
ID Sync Error	00	00	0E
Data Sync Error	Obsolete		
Sync Loss Error	Obsolete		
Data Compare Error	00	00	11
Recoverable Error	00	00	12
Unrecoverable Error	00	00	13
Bad Block Error	Obsolete		
Direct Errors			
Manually Entered Error	Obsolete		
Not Selected Error	0A	04	18
Not Ready Error	0B	04	05
Aborted Command	44	04	0B
DRQ Error	2F	04	14
IRQ Error	3E	04	15
Command Timeout	24	04	17
Recal Error	10	04	05
Parity IN Error	04	04	19
Parity OUT Error	04	04	1A

EL

Error location variable holds the last location of a data compare error.

ET

Legacy error type.

FA-FH

These eight variables allow custom error handling. The variables FA through FH can be used to facilitate custom failure actions, specific to certain error codes.

USAGE: SET VARIABLE v=FA (x)n=wwxxyyzz

ww = SCSI SENSE KEY (FF is a wild card)

xx = SCSI ADDITIONAL SENSE CODE (FF is a wild card)

yy = SCSI ADDITIONAL SENSE CODE QUALIFIER (FF is a wild card)

zz = Action Code:

F0 = Ignore

F8 = Fail Immediately

F9 = Gosub Command Complete Timeout subroutine.

FB = Handle and log as a READ error.

FC = Handle and log as a SEEK error.

FD = Treat as a NOT READY error in TEST SPIN UP macro.

Examples:

SET VARIABLE v=FA (x)n=0B4700FB * Log all 0B/47/00 errors as READ errors.

SET VARIABLE v=FB (x)n=0B47FFFB * Log all 0B/47 errors (ANY SENSE QUAL) as READ errors.

FP

File Pointer variable. This variable sets the file pointer (offset) for file upload and download operations.

FS

Fibre channel speed variable.

FU

False error generation used for debugging on-error-gosub constructs and check error handler functionality.

HD

Current head number (IDE).

HI

Host interval - This variable may be used to reduce the network tasks from interfering with the testing tasks. The value in "HI" is in PC ticks and it is the amount of time (in ticks; 1/18 sec) before the next network status page update is done. The default value is 9. It is not recommended to use a large value for "HI" because it will cause the port to go offline. A value up to 128 (80h) has been tested on single port systems. You must set this using the "Set Variable" command. This value will be reset when a new script is started.

HN

Minimum head number.

HX

Maximum head number.

IS

This 1 byte variable contains the IOC status from the SCSI host adapter.

L0 (L1-L4)

This variable contains the current LBA for SCSI and ATAPI interfaces.

LA,LB,LC

These 2 byte variables contain the loop count for LOOP 1, LOOP 2, and LOOP 3 respectively.

LE

This variable contains the number of Log Entries (error list entries).

LH

Link Status C0 Link Failure count (Parity Test Result, Fibre Channel Only).

LI

Link Status C1 Loss of Sync count (Parity Test Result, Fibre Channel Only)

LJ

Link Status C2 loss of signal count (Parity Test Result, Fibre Channel Only).

LK

Link Status C3 primitive sequence error (Parity Test Result, Fibre Channel Only).

LL

Link Status C4 Invalid xmit word(Parity Test Result, Fibre Channel Only).

LM

Link Status C5 Invalid CRC count (Parity Test Result, Fibre Channel Only).

LN

Current script line number.

LT

Timed loop value.

LX

Logical LBA's per cylinder.

MM (16 char string)

The MM variable contains the Manufacturer Model number string. This string can be up to 16 characters in length. The string is normally set by the CONFIGURE PORT macro.

MN

This variable contains the Minimum LBA value. The CONFIGURE PORT macro sets this value to zero but the value can be changed at any time in the script by setting this variable.

MO

Min LBA Ext (ATA 48 bit LBA mode only).

MP

This variable contains the pass count for macro commands.

MR

Middle LBA.

MS

Step mode variable. This variable controls the state of the script single step mode. MS can have three valid values: 0 = RUN mode (default). 1 = MACRO step mode. 2 = COMMAND step mode. Any other value may produce undefined results.

MX

This variable contains the Maximum LBA value for SCSI and ATAPI interfaces. This value is normally set by the CONFIGURE PORT macro but can be set directly by using the "*SET VARIABLE*" command.

MY

Max LBA Ext (ATA 48 bit LBA mode only).

NA

Port MAC address (network address).

NB

Number of blocks between seeks.

NC

Number of commands.

NH

Number of heads.

NL

Number of cylinders.

NN

SCSI SDTR negotiated value. Byte 1 = offset, 2 = period, 3 = opcode, 4 = Results.

NO

This variable contains the current cylinder value if operating in a physical (CHS) mode.

NS

Contains the number of sectors per track. This is normally calculated by the CONFIGURE PORT macro.

OC

Open / Close variable for ATAPI (tray open).

OF

Resulted Option/Feature set after PPR negotiation (SCSI)

OT

Oven temperature (degrees C).

OU

Oven humidity (%).

P0

Head value (4 bytes).

PC

Preceding command (ATA).

PM

Model number of the D.U.T. – up to 12 characters.

PN

This variable contains the current port number.

PO

General purpose 4 byte variable.

PT

Port Type variable. This variable is set by the CLIENT software to identify the interface of the device being tested.

PV

Contains the firmware version number of the NPM power card (4 bytes).

QA-QE

These five variables hold the device serial number. Since each variable can hold 4 bytes, the serial number string is broken up into 4 byte segments.

RM

Reminder of Division Operation.

RN

Minimum random transfer length. Used to control the lower limit of the SET RANDOM VARIABLE command. See also example in **RX**.

R0

Currently generated random number. See also example in **RX**.

RR

UDMA CCTO retry command timeout time.

RS

Random Seed variable. This variable contains the SEED for the random number generator. The RS variable, when set to a value, will cause the random number generator to generate pseudo-random number based on the value in RS. Also, some macro commands such as WRITE RANDOM TRK will utilize the random number generator to pick the random values for cylinder and head (or LBA). If the RS variable is set prior to the macro command, the generator will generate the same sequence of pseudo-random numbers.

RT

Compare Result.

RX

Maximum random transfer length. Used to control the upper limit of the SET RANDOM VARIABLE command.

EXAMPLE:

```
<014> DATA COMPARES (0=on, 1=off) =1
<015> SET VARIABLE v=RN  (x)n=00000000
<016> SET VARIABLE v=RX  (x)n=000000FF
<017> { Start Loop #1  loops=50
<018> SET RANDOM VARIABLE v=R0
<019> MOVE VAR  s=R0 TO VAR d=X0  ; generate a new random xfer length
<020> *
<021> READ RANDOM TRACK pass=1   ; read with random xfer length
<022> *
<023> } End Loop #1
```

SA

SCSI TUR (test unit ready) delay. This variable allows the user to select the delay in 0.1 second increments between subsequent execution of the REQUEST SENSE command. The command is repeatedly issued during a TEST SPIN UP where the function is waiting for drive ready after power is applied.

SB

Test Spinup Limit.

SC

Seek counter.

SD

Spin up Delay variable. This variable inserts a delay in 0.1 second increments between POWER ON and the start of TEST UNIT READY commands (SCSI) during the TEST SPIN UP macro.

SL,SH

Command time total. SL = low 32 bits, SH = high 32 bits.

SK

Contains the SENSE KEY from the last REQUEST SENSE command (SCSI, ATAPI).

SN (string)

16-byte variable used to store the device serial number. Serial number is usually acquired by the CONFIGURE PORT macro.

SO

Serial number offset variable is used to parse the serial number string for file naming, etc.

USAGE:

SO byte	Description
0	Offset into the 20-byte S/N string.
1	Number of characters used (size).
2	Manual data length. When this value is nonzero, a composite serial number filename is created using data from the manual serial number entry and the drive S/N retrieved from the INQUIRY or MODE SENSE commands.
3	Bit Description 0 Inhibit srch INQY data S/N 1 Inhibit srch MDSNS data S/N 2 Get S/N through VPD page 80h 3 Use CCS INQY VPD pages 4 Get S/N through VPD page E0h

SP

Spinup time variable. Results are in 0.1 second increments.

SS

Sector size variable is usually set by the CONFIGURE PORT macro but has a default value of 512.

ST

Error status variable.

SX

LBA's per cylinder.

SY

LBA's per track.

TE

Sense dump threshold. Default = 255.

TH

Total Blocks Transferred HIGH contains the 4 MS bytes of the number of blocks (sectors) read. Variable TL contains the LS 4 bytes of this number.

TL

Total Blocks Transferred LOW contains the 4 Least Significant bytes of the number of blocks (sectors) read. Variable TH contains the Most Significant 4 bytes of this number.

TN

Contains the Target ID for SCSI and ATAPI devices. This variable is normally set by the SET TARGET ID command.

TP

Current temperature reading.

TT

Total Open/Close Tray count.

TU

CPU temperature – Pegasus only.

TZ

Mother board temperature - Socket 370 pegasus system only.

UT

User Timeout(usec).

VA, VB

Variables "VA" & "VB" - first 0-4 and 5-7 bytes of the revision field in a ATAPI Identify command parameter. This only gets loaded during ATAPI Configure Port.

VC

Channel 0 (5V) variable. The high word contains the voltage in millivolts and the low word contains the current in milliamps.

VD

Channel 1 (12V) variable. The high word contains the voltage in millivolts and the low word contains the current in milliamps.

VO

This variable contains 32 bits that are used as program option control bits. This variable has different meanings depending on the particular hardware interface being used. The meanings are as follows:

IDE/ATAPI/SCSI:

VO Byte	Bit	Description
0	0	Use extended address in var IO for RD/WR register ops (PCMCIA). **IDE**
	0	Enable IDE read ahead. **ATAPI**
	0	Get CHS from Sense offset 24-27. **SCSI**
	1	Measure Command overhead. **IDE/SCSI**
	1	Ignore UA on a START DISK command. **ATAPI**
	2	Run controller chip in polling mode (non-interrupt mode). **IDE**
	2	Enable inch worm Seq RD/WR. **SCSI/ATAPI**
	3	Don't fail on SET HOST DMA. **IDE**
	3	Translate defect (page 40) **ATAPI/SCSI**
	4	Perform a *read* instead of *seek* during TEST SPIN UP command. **IDE/ATAPI**
	4	Randomize transfer length. **SCSI**
	5	Inhibit SEEK or READ cmd in TEST DRIVE SPINUP. **IDE/ ATAPI**
	5	Track Squeeze on Osc Write. **SCSI**
	6	Disable timeout recovery. **IDE**

	6	Use 10 byte CDB's. **SCSI/ATAPI**
	7	Use WRITE VERIFY command instead of WRITE. **IDE/ATAPI/SCSI**
1	8	Don't update Task File Registers on Timeout type Errors. **IDE**
	8	Skip Reading TOC. **ATAPI**
	8	Use SCSI Page 2 format. **SCSI**
	9	In Test Spin-up, wait abort limit - SD duration for BSY for SRST. **IDE/ATAPI**
	9	Get CHS from sense offset 19-22. **SCSI**
	10	Use value in Feature Register as num of blocks to transfer (VU). **IDE/ATAPI**
	10	Don't issue START cmd in TEST DRIVE SPINUP. **SCSI**
	11	Don't fail on detection of invalid status. **IDE/ATAPI**
	11	Use FUA bit in commands. **SCSI**
	12	Use variable 'XC' for sector count. **IDE/ATAPI**
	12	Request QAS during Fast-160/320 negotiation. **SCSI**
	13	Spare. **IDE/ATAPI**
	13	Clear timer before start command. **SCSI**
	14	Override 15% limitation of power margin circuit. **IDE/ATAPI/SCSI**
	15	Use SEEK retry algorithm. **IDE/ATAPI/SCSI**
2	16	Don't sort defect data. **IDE/ATAPI**
	16	Add command time to total time. **SCSI**
	17	Use Zone table on VU commands. **IDE/ATAPI**
	17	Set DPO bit in commands. **SCSI**
	18	Limit sec/trk length on zoned VU commands. **IDE/ATAPI**
	18	Inhibit S/N search in INQUIRY command. **SCSI**
	19	Wait for IRQ before first transfer on data out. **IDE/ATAPI**
	19	Inhibit S/N search in Mode Sense command. **SCSI**
	20	Causes the data transfer to halt on data compare. This allows the contents of the track buffer to remain intact following a data compare error. **IDE/ATAPI**
	20	Single Step on selection timeout. **SCSI**
	21	Check timed loop on every I/O – exit macro if done. **IDE/ATAPI/SCSI**
	22	Log DRQ, IRQ, and ABORT errors in defect list. **IDE/ATAPI**
	22	Single Step on Command timeout. **SCSI**
	23	Get CHS from Sense offset 8-11. **SCSI**
	24	Disable track buffer rebuild on data compare. **IDE/ATAPI**
3	24	Manual data phase (long file transfer). **SCSI**
	25	Save track buffer but no pause. **IDE/ATAPI**

25 End macro on TH,TL limits. **SCSI**

26 Macro exit on a timed loop. **IDE/ATAPI**

26 End macro on BH, BL limits. **SCSI**

27 Use Read Verify instead of Read command. **IDE/ATAPI/SCSI**

28 Allow Test Spinup Failures to use gosub. **IDE/ATAPI/SCSI**

29 Special transfer handling: Sets chip for max transfer. **IDE/ATAPI/SCSI**

30 Spare. **IDE/ATAPI**

30 Disable fail on Request Sense check. **SCSI**

31 If CRC error, retry one time. If next transfer is OK, continue. **IDE/ATAPI**

31 Enable sense dump. **SCSI**

WM

This variable indicates SCSI wide mode if = 1.

X0

Transfer length variable contains the number of blocks (sectors) that are to be transferred during a write or read operation.

YA

Alternate Status Register

YB

Device Control Register.

YC

Sector Count Register.

YD

Device Select/Head Register.

YE

Error Register.

YF

Feature Register.

YH

Cylinder High Register.

YL

Cylinder Low Register.

YN

Sector Number Register.

YS

Status Register.

YO

Command Register

6.4 ABORT LIMITS

For every test generated in the Flexstar test editor, there is an associated set of abort limit values. The abort limits are set on the right hand side of the editing window. Make any modifications by clicking on the abort limit field to be modified.

Abort limits are checked constantly during the test. If any limit is exceeded, the test will abort immediately without finishing the test.

NOTES:

1) When a new test script is created, all abort limit values will be set to zero (default). Note also that each abort limit is a 16-bit value and therefore has a maximum value of 65535. Some abort limit values are Boolean – 0 = false, > 0 = true.

2) Most references to soft errors are not valid at the time of this writing. Soft errors were important in the legacy system with older, less intelligent, interfaces. Most errors are treated as hard errors since the drive internally handles retries and applies error correction to detected errors. Soft errors are simply non-repeatable errors that have not exceeded the CREDENCE threshold. If an error reoccurs beyond the credence level, the error is considered a hard error.

Below is a listing of the abort limits available in the Flexstar software.

LIMIT NAME	DEFAULT VALUE	MAX VALUE	TYPICAL VALUE	DESCRIPTION
MAX Test Time (Mins)	0	65535	0	Sets the maximum time in minutes for the test. The value is checked only during IMAGE FILE updates, which occur between test commands. Therefore, if the MAX Test Time is exceeded while a program line is being executed, the test will not abort until that command has completed. A value of 0 disables time checking.
MAX Ready Errors	0	999	0	Sets the number of Drive Not Ready errors that can occur before aborting the test. If set to 2, the third error will abort the test.
MAX Select Errors	0	999	0	Sets the number of Drive Not Selected errors that can occur before aborting the test.
MAX Write Faults	0	999	0	Sets the number of Write Fault errors that can occur before aborting the test.
MAX Recalibrate Errs	0	999	0	Sets the number of Recalibrate errors that can occur before aborting the test.

LIMIT NAME	DEFAULT VALUE	MAX VALUE	TYPICAL VALUE	DESCRIPTION
MAX Seek Errors	0	999	0	Sets the number of Seek errors that can occur before aborting the test.
MAX Index T.O. Errs	0	999	0	Sets the number of Index Time-out errors that can occur before aborting the test.
MAX Sync Errors	0	999	0	Sets the number of combined ID sync and DATA sync errors that can occur before aborting the test.
MAX Compare Errors	0	999	0	Sets the number of Compare errors that can occur before aborting the test.
MAX Command Timeouts	0	999	1	Sets the number of Command Timeout errors that can occur before aborting the test.
HARD Error Credence	0	999	2	Sets the number of times the drive will attempt to read an error before it is listed as a HARD error. Do not set this limit to 0 (illegal value).
SEEK Error Credence	0	999	0	Sets the number of times the drive will attempt to seek to the desired location before the tester issues a seek error.
MAX Hard Errs/Head	0	999	20	Sets the number of hard errors that can occur on any given head.
MAX Errors Per Track	0	65535	10	Sets the maximum allowable errors that can occur on any given track
MAX Hard Errs/Drive	0	65535	100	Sets the maximum allowable number of hard errors on the entire drive.
MAX New Hard Errors	0	999	10	Sets the number of new hard errors that can occur on the entire drive. New hard errors are defined as any hard error that was not read from the existing defect map on the drive.
MAX Soft Errs/Head	0	65535	2	Sets the number of soft errors that can occur on any given head.
MAX Soft Errs/Drive	0	65535	10	Sets the maximum allowable number of soft errors on the entire drive.
MAX RNF / AMNF Errors	0	999	0	Sets the number of Record Not Found errors (address mark) that can accumulate on the entire drive.

LIMIT VALUE	DEFAULT VALUE	MAX VALUE	TYPICAL VALUE	DESCRIPTION
MAX CRC Errors	0	999	UNK	Sets the number of CRC errors that can occur before aborting the test.

PRN Soft Errors (0=NO)	0	1	1	A toggle to control whether or not soft errors will be printed from the defect map listing in the test report.
DEF Free CYL Start 1	0	9999	0	Sets the beginning cylinder of defect free zone #1 on the drive. Entering a number greater than the maximum number of cylinders on the drive will disable this feature.
DEF Free CYL End 1	0	9999	0	Sets the ending cylinder of defect free zone #1 on the drive. The test will abort if defects are found within the defect free zone.
DEF Free CYL Start 2	0	9999	0	Sets the ending cylinder of defect free zone #2 on the drive. Entering a number greater than the maximum number of cylinders on the drive will disable this feature.
DEF Free CYL End 2	0	9999	0	Sets the ending cylinder of defect free zone #2 on the drive. The test will abort if defects are found within the defect free zone.
INDEX Low (x10ms)	0	65535	0	Sets the lowest limit for index timing on the index pulse (based on the highest speed of rotation).
INDEX High (x10ms)	0	65535	0	Sets the highest limit for index timing on the index pulse (based on the lowest speed of rotation).
MIN 1 Tk Seek (*.1ms)	0	65535	UNK	Sets the minimum time for a 1 track seek in the TEST SEEK TIMING command.
MAX 1 Tk Seek (*.1ms)	0	65535	UNK	Sets the maximum time for a 1 track seek in the TEST SEEK TIMING command.
MIN 1/3 Seek (*.1ms)	0	65535	UNK	Sets the minimum time for a 1/3 track seek in the TEST SEEK TIMING command.
MAX 1/3 Seek (*.1ms)	0	65535	UNK	Sets the maximum time for a 1/3 track seek in the TEST SEEK TIMING command.
MIN FS Seek (*.1ms)	0	65535	UNK	Sets the minimum time for a full stroke seek in the Test SEEK Timing command.
MAX FS seek (*.1ms)	0	65535	UNK	Sets the maximum time for a full stroke seek in the TEST SEEK TIMING command.
MIN Avg Seek (*.1ms)	0	65535	UNK	Sets the minimum time measurement for the AVERAGE SEEK TIMING command.

MAX Avg. Seek (*.1ms)	0	65535	UNK	Sets the maximum time measurement for the AVERAGE SEEK TIMING command.
MIN Osc Seek (.*1ms)	0	65535	UNK	Sets the minimum time measurement for the OSCILLATING SEEK TIME command.
MAX Osc Seek (*1ms)	0	65535	UNK	Sets the maximum time measurement for the OSCILLATING SEEK TIMING command.
MAX Spin Up (*sec)	0	999	45	Sets the maximum allowable time for the drive to come ready after power on when the TEST SPIN UP command is used. Works in conjunction with the FlexStar power margining boards only. If no power margin board is present, value should be set to 0.
MIN 5 v Cur (*.125a)	0	999	UNK	Checks if the minimum 5V current to the device is less than this value. Works in conjunction with the FlexStar power margining boards only.
NOM 5v Cur (*.125a)	0	999	UNK	Checks if the nominal (averaged) 5V current is exceeded. Works in conjunction with FlexStar power margining boards only.
MAX 5v Cur (.*125a)	0	999	UNK	Checks if the maximum 5V current to the device is exceeded. Works in conjunction with FlexStar power margining boards only.
MIN 12v Cur (*.125a)	0	999	UNK	Checks if the minimum 12V current to the device is less than this value. Works in conjunction with FlexStar power margining boards only.
NOM 12v Cur (*.125a)	0	999	UNK	Checks if the nominal (averaged) 12V current is exceeded. Works in conjunction with the FlexStar power margining boards only.
MAX 12v Cur (*.125a)	0	999	UNK	Checks if the maximum 12V current to the device is exceeded. Works in conjunction with FlexStar power margining boards only.
MAX Data Overrun Errs	0	99	0	Sets Maximum Data Overrun Errors allowable. These occur when the DRQ bit is set at the end of a data transfer (IDE).
MAX No Interrupt Errs	0	99	0	Sets Maximum No Interrupt Errors allowable. These occur when the IRQ is not asserted at the beginning of a data transfer (IDE).
MAX Drive Abort Cmds	0	99	0	Sets maximum drive aborted commands. These occur when the drive reports that a command was aborted (IDE).

OVERLIMIT Temp (deg)	0	999	0	Sets the maximum temperature allowed before aborting the test.
UNDERLIMIT Temp (deg)	0	999	0	Sets the minimum temperature allowed before aborting the test.
BYTES Per Sector	0	65535	0	Not Used.
LANDING Cylinder	0	65535	0	Sets the landing zone for the heads to seek to once the test is completed or aborted, before the port turns power off to the drive. If set to 9999, the LAND HEADS command is ignored.
SCRATCH Width (bytes)	0	999	0	Not implemented.
SCRATCH Length (Cyls)	0	65535	0	Not implemented.
SCRATCH Qualif (Cyls)	0	65535	0	Not implemented.
CUSTOM SPAWN Arg-1	0	99	0	Not Used.
ABORT Pass Flag (1=on)	0	1	0	Not Used.
MIN Soft Rate E-n	0	999	0	Sets the minimum error rate allowed for soft errors.
MIN Hard Rate E-n	0	999	0	Sets the minimum error rate allowed for hard errors.
MIN Seek Rate E-n	0	999	0	Sets the minimum seek rate allowed.
Hard Err Rate Flag (1=on)	0	1	0	Not Used.
MIN Ch0 Cur (ma)	0	65535	UNK	Sets minimum current allowed on Channel 0
NOM Ch0 (ma)	0	65535	UNK	Sets nominal current on Channel 0.
MAX Ch0 (ma)	0	65535	UNK	Sets maximum current allowed on Channel 0.
MIN Ch1 (ma)	0	65535	UNK	Sets minimum current allowed on Channel 1.
NOM Ch1 Cur (ma)	0	65535	UNK	Sets nominal current on Channel 1.

NOTE 1: UNK means that the results are dependent on the device under test and cannot be recommended here.

7.0 Report Generator Description

The text REPORT files are created from a script driven report generator that is tightly integrated with the Orion Host software. This method of report generation allows maximum flexibility toward user customization.

The basic function of the report generator is to extract information from one or more database files and create a formatted text output based on the contents of a script file. The script file must have the file extension .ORP (Orion Report Program).

7.1 Report Generator Script Language Description

Orion report generator script is written in simple ASCII text. The Orion Host has a built-in report script editor, which allows editing and also allows viewing of the output based on a selected database file. This gives the user the ability to test scripts on database files and verify correct formatting of the report output without having to re-run a test each time the script is modified.

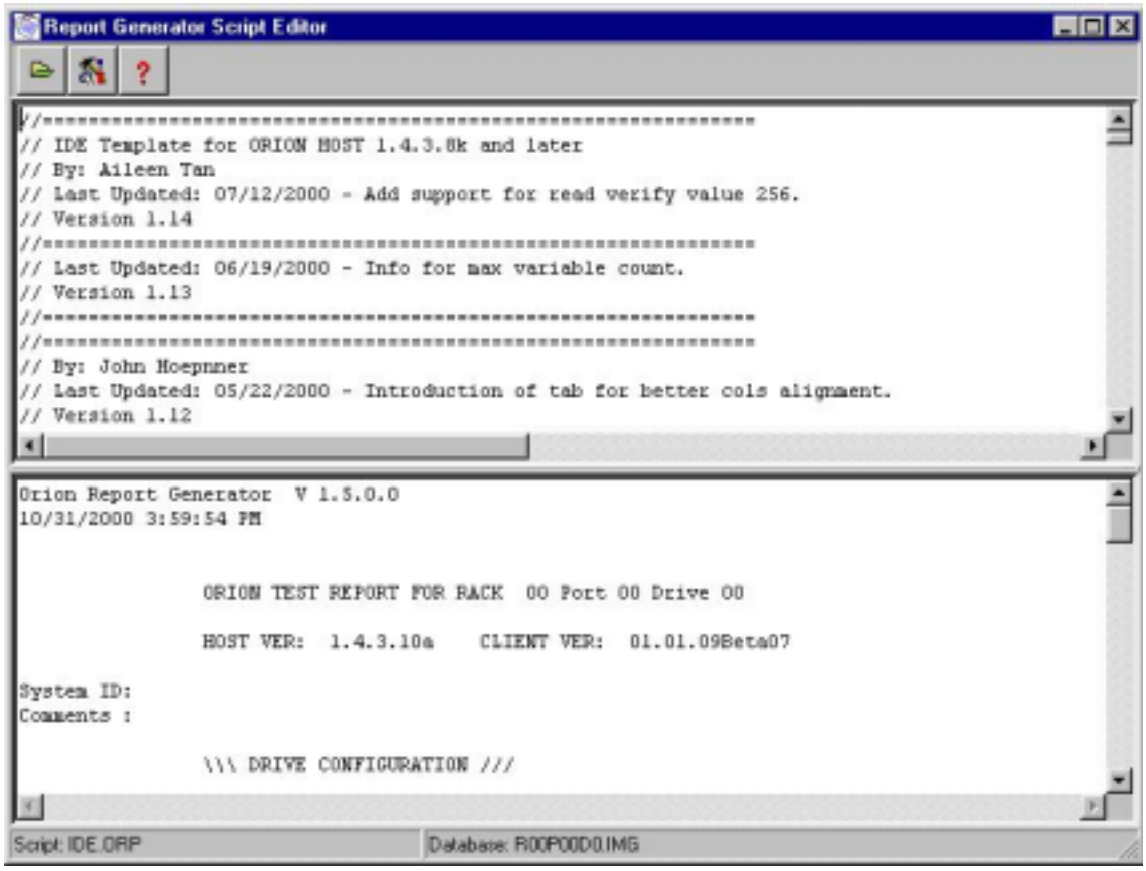

Report Generator Script Editor
//==
// IDE Template for ORION HOST 1.4.3.8k and later
// By: Aileen Tan
// Last Updated: 07/12/2000 - Add support for read verify value 256.
// Version 1.14
//==
// Last Updated: 06/19/2000 - Info for max variable count.
// Version 1.13
//==
//==
// By: John Hoepnner
// Last Updated: 05/22/2000 - Introduction of tab for better cols alignment.
// Version 1.12
Orion Report Generator V 1.5.0.0
10/31/2000 3:59:54 PM
ORION TEST REPORT FOR RACK 00 Port 00 Drive 00
HOST VER: 1.4.3.10a CLIENT VER: 01.01.09Beta07
System ID:
Comments :
\\\ DRIVE CONFIGURATION ///
Script: IDE.ORP
Database: R00P00D0.IMG

Figure 25 - Report Generator Script Editor

7.1.1 Orion Report Generator Command List

The Orion report generator is a script driven text report generator based on the standard 'C' printf statement. In addition, several commands are included to establish behavior and provide conditional program flow.

The Orion report generator provides the report formatting information by extracting data from a 'database' file and organizes the report output based on a Format string provided in the script. The result is written to a text output file.

Each command begins with @ symbol followed by one or more characters which form a command mnemonic. The mnemonic is interpreted by a script interpreter, which is built into the Orion Host software. A list of the commands follows:

@F - Use Database File

The File command gets a database file at run time. This database file is used until another @F command is encountered.

<USAGE> : @F <database filename>

<EXAMPLE> : @F R00P00D0.IMG

It is also possible to modify the file extension of the current database file by using the following syntax:

@F *.<ext>

This is useful for accessing two files of the same name but with different file extensions.

@P - Print Command

The Print command prints a formatted string and argument list in a form very similar to the 'C' printf function. The format string must be enclosed in quotes and commas must separate all optional argument fields that follow. See also *Format String* below.

The @P command mnemonic must be the first two non-white space characters of the script line.

<USAGE> : @P <format string> , arg1, arg2, etc.

<EXAMPLE> : @P "Orion Test Report %s\n", _V // prints title with version#

@V - Set Variable Command

Report generator script variables must be defined prior to their use in the script. A good practice is to define all variables at the beginning of the program, which provides good structure and can also improve the performance of the script engine.

The @V command initializes a new variable and optionally sets its initial value.

<USAGE> : `@V <type> <varname> = <value>`

<EXAMPLE> : `@V int newvar = 100`

The description for each field is as follows:

<type> This describes the data type of the new variable. Valid types are:

* INT - 32 bit integer data type.

* STR - String variable of 256 characters max.

* FLT - Floating-point variable.

@AV - Set String ARRAY Command

The @AV command allows the use of array variables to convert enumerated values (error codes, etc.) into string values. Loading the strings from a text file initializes the array variable. An ordinal index value can then be used to access the array.

<USAGE> : @AV <type> <varname> = <filename>

<EXAMPLE> : @AV str strArray = ERRCODE.TXT

The string array can be used in the @P print command as follows:

```
@P "Value of the string array is %s\n", strArray[ERR_CODE]
```

@L - Loop Command

The @L command provides a LOOP control method within the script. The @L argument is the number of loops to complete. Note that a loop value of 0 (zero) will not perform any part of the loop construct within the enclosing braces. The <loop count> can either be a constant or a variable of type INT.

<USAGE> : `@L <loop count>`

```
{  start of loop

  do something ...

}  end of loop
```

<EXAMPLE> : `@L 100`

```
{

    @P "Hi There!\n"

}
```

{ - Starting Brace

The starting brace symbol must be the first non-white space character in the script line. The starting brace designates the beginning of a BLOCK of script lines that are a part of a LOOP command. No arguments are required.

<USAGE> : `{`

} - Ending Brace

The ending brace symbol must be the first non-white space character in the script line. The ending brace designates the END of a BLOCK of script lines that are a part of a LOOP command. No arguments are required.

<USAGE> : }

@IF - If Condition Command

The 'IF' command controls script execution by executing a section of script statements if the statement argument evaluates TRUE. The 'IF' statement is either followed by a single script statement or enclosing braces to indicate that a larger block of script will be conditionally executed.

<USAGE> : @IF <variable> <logical operation> <constant or variable>

```
{
 execute if TRUE...
}
```

<EXAMPLE> : `@IF MyVar <= 5`

```
{
 @P "MyVar was <= 5!"
}
```

@ELSE - Else Condition Command

The @ELSE statement is always used with the @IF statement to provide an alternate decision path to the IF decision. The @ELSE must always follow the @IF and will execute it's section of script if the @IF statement evaluates FALSE.

<USAGE> : @ELSE

```
{
 Execute if the @IF evaluated FALSE
}
```

<EXAMPLE>:@IF MyVar <= 5

```
{
 @P "MyVar was <= 5!"
}
  @ELSE
{
 @P "MyVar was > 5!"
}
```

@TAB – Set Tab Position

The @TAB command sets the tab boundaries for the %t format string specifier. Example:

@TAB = 4 (tab on column multiples of 4)

Assign Operation (=)

Variables can be assigned values anywhere in the script after they have been defined by the @V command.

<USAGE>**:** <var> = <operation>

<EXAMPLE>: NewVar = NewVar + 1 // adds 1 to NewVar

```
   NewVar = 5        // set NewVar value equal to 5
```

Comments

Comment fields can be placed anywhere in the script. They must start with either a double slash '//' or a semicolon ';'. Comments are ignored and are only included in the script for documentation.

Format String

Format strings used by the @P command contain two types of objects -- literal characters and format specifiers. Literal characters are copied verbatim to the resulting string. Format specifiers fetch arguments from the argument list and apply formatting to them.

Format specifiers have the following form:

"%" [width] ["." prec] type

A format specifier begins with a % character. After the % come the following, in this order:

An optional width specifier, [width]

An optional precision specifier, ["." prec]

The conversion type character, **type**

The following table summarizes the possible values for type:

d Decimal. The argument must be an integer value. The value is converted to a string of decimal digits. If the format string contains a precision specifier, it indicates that the resulting string must contain at least the specified number of digits; if the value has less digits, the resulting string is left-padded with zeros.

f Float. The argument must be a floating-point value. The value is converted to a string of the form "-ddd.ddd...". The resulting string starts with a minus sign if the number is negative.The number of digits after the decimal point is given by the precision specifier in the format string—a default of 2 decimal digits is assumed if no precision specifier is present.

s String. The argument must be a string variable. The string is inserted in place of the format specifier. The precision specifier, if present in the format string, specifies the maximum length of the resulting string. If the argument is a string that is longer than this maximum, the string is truncated.

x Hexadecimal. The argument must be an integer value. The value is converted to a string of hexadecimal digits. If the format string contains a precision specifier, it indicates that the resulting string must contain at least the specified number of digits; if the value has fewer digits, the resulting string is left-padded with zeros.

t Tabbed position. Forces text to start on a tabbed boundary (column). The tab column is specified using the @TAB command. The tab format specifier can contain an optional decimal value which specifies an absolute column number to begin on. Examples:

```
@V STR name

name = “hi there”

@P “%t%s”, name;          (places “hi there” on the next tab boundary)

@P “%20t%s”, name;        (places “hi there” on absolute column 20)
```

Conversion characters may be specified in uppercase as well as in lowercase—both produce the same results.

A width specifier sets the minimum field width for a conversion. If the resulting string is shorter than the minimum field width, it is padded with blanks to increase the field width. The default is to right-justify the result by adding blanks in front of the value, but if the format specifier contains a left-justification indicator (a "-" character preceding the width specifier), the result is left-justified by adding blanks after the value.

7.1.2 Keyword List

Certain words are reserved and should not be used except as special keywords. Keywords that begin with an underscore ("_") are macros used with the @P command. The remaining keywords are used to identify variable data types.

_D – Date + Time String

The _D keyword provides a string that represents the current date and time in the format: MM/DD/YYYY HR:MIN:SEC in 24 hour time.

```
<EXAMPLE>:@P "Date & Time=%s\n", _D   // prints the current date and time
```

_V - Revision Level String

The _V keyword, when used as a string constant, represents the current revision level of the 'REPGEN.DLL'. This string will appear as follows:

'V 1.01' where the digits from left to right have the meaning:

* Major revision level - not guaranteed to be down level compatible.

* Minor revision level - compatible with previous versions of minor revisions but includes new features / enhancements.

_DB - Extract BYTE

The _DB keyword is used to extract a BYTE (8 bits) of data from the database file.

```
<EXAMPLE>:@P "Count = %d\n", _DB[20]  // extract value from DBFILE[20]
```

_DW - Extract WORD

The _DW keyword is used to extract a WORD (16 bits) of data from the database file.

```
<EXAMPLE>:@P "Count = %d\n", _DW[21]  // extract value from DBFILE[21]
```

_DD - Extract DWORD

The _DD keyword is used to extract a DOUBLE WORD (32 bits) of data from the database file.

```
<EXAMPLE>:@P "Count = %d\n", _DD[22]  // extract value from DBFILE[22]
```

_DS - Extract STRING

The _DS keyword is used to extract a STRING from the database file. Note that two parameters are required for string extraction: OFFSET & LENGTH.

<EXAMPLE>: `@P "S/N = %s\n", _DS[10,8]  // extract STRING from DBFILE[10,8]`

INT - Define INT Type

The INT keyword is used in variable assignments to identify the data type of the new variable as INT (32 bit).

<EXAMPLE>: `INT myint = 0     // creates an integer variable called “myint”`

STR - Define String Type

The STR keyword is used in variable assignments to identify the data type of the new variable as a STRING variable. String variables may be up to 256 characters in length.

<EXAMPLE>: `STR mystr      // creates a string variable called “mystr”`

FLT - Define Floating Point Type

The FLT keyword is used in variable assignments to identify the data type as FLOATING POINT.

<EXAMPLE>: `INT myfloat = 0      // creates an float variable called “myfloat”`

8.0 Oven Control Description

The Orion Host software has the ability to control and synchronize device test scripts with oven control. A Flexstar burn-in system or environmental chamber includes a temperature controller that is controlled from the server via a RS-232 or RS-485 connection.

Oven control is accomplished through a script language, which can be created and edited via a special oven script editor provided.

The oven script language is capable of setting event outputs, controlling oven ramp rates, and allowing simple script program looping.

The oven script can also respond to events created by running device test scripts. This allows the oven to be synchronously mastered.

8.1 Oven Script Editor Operation

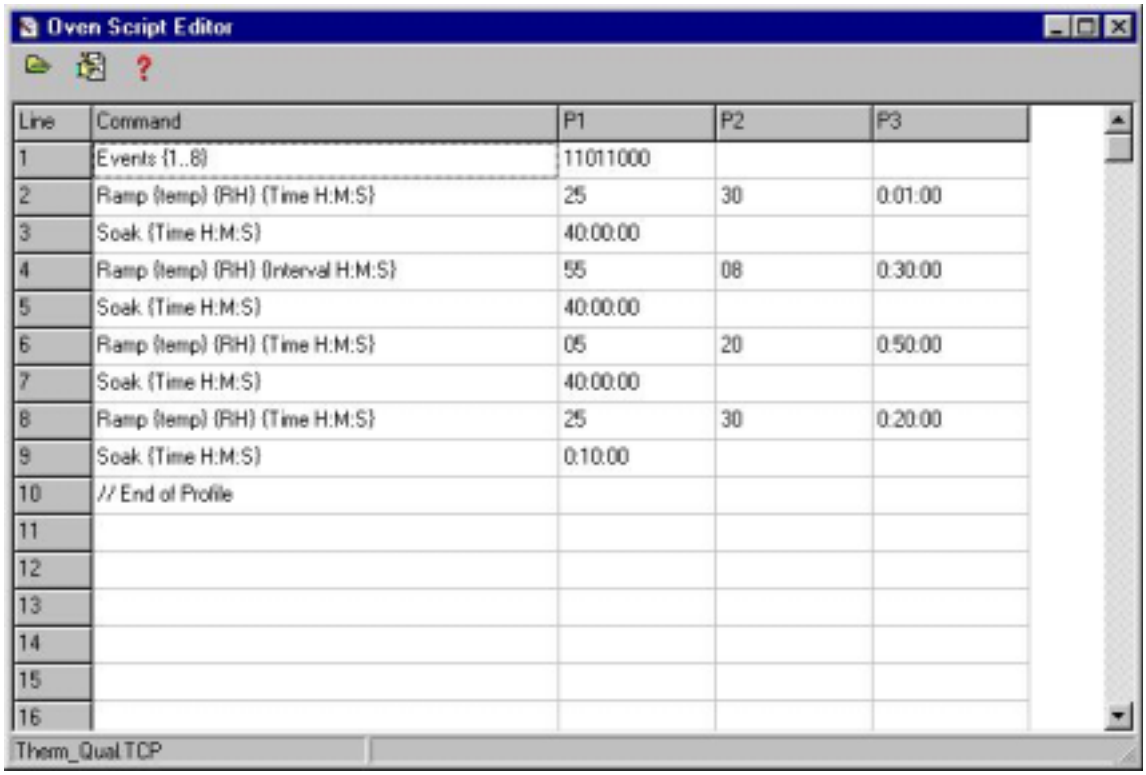


Figure 26 - Oven Script Editor

The oven script editor is activated by selecting the 'Editors' button on the Host toolbar. Select 'Oven Program Editor' from the drop-down menu. This will launch the Oven Editor utility.

8.1.1 Opening an Existing Script File

Click on the File Menu toolbar button and select Open. This will launch a file open dialog. Select the desired TCP file and click Open.

8.1.2 Saving a Script File

Click on the File Menu toolbar button and select Save or Save As. If Save As was selected, this will launch a file save dialog. Type in the name of the file to be saved and click Save.

8.1.3 Creating a New Script

Delete the contents of the exsiting program or close the Oven Editor and re-start it.

8.1.4 Adding New Commands

A new script command can be added by selecting a script line and then clicking on the Edit Menu button on the toolbar. This will display a dropdown menu. Click on “Insert New Command” which will display a submenu with all possible script commands. Select the desired command and this command will then be inserted above the currently selected script line.

8.1.5 Deleting Commands

Commands can be deleted by selecting the script line to be deleted and then clicking the Edit Menu toolbar button. Click on “Delete Command” in the dropdown menu to remove the command.

8.1.6 Replacing Commands

Commands can be replaced (overwritten) by using the same procedure as adding new commands except select the “Replace Current Command” menu item. This will replace the current command with the new command.

8.1.7 Checking Program Syntax

Program syntax can be checked using the Check Program function under the Edit Menu toolbar button. This function informs the user of possible syntax errors in the script.

The dropdown menu that is accessed via the Edit Menu toolbar button can also be accessed by clicking the RIGHT mouse buton over the oven script editor.

8.2 Oven Script Language Description

There are eight possible oven script language commands:

8.2.1 Autostart

The autostart function allows program start to be initiated when the specified date and time are achieved. The arguments are date (mm/dd/yyyy) and time (hh:mm:ss).

8.2.2 Comment

Comments are not executed and are present to describe program functionality. Comments always begin with a double slash and are ignored at run time.

8.2.3 Events

Temperature controller output Events are controlled by this command. The typical controller has 8 event outputs. Each event output (1 -> 8) is turned on by setting its value to '1' or off by setting it's value to '0'.

8.2.4 LoopA, LoopB, LoopC

There are three independent loop constructs in the script language. These loop functions provide simple loop capabilty. The loop arguments are a destination line number and an iteration count.

If the destination line number is less than the line number of the loop command, the command will jump backwards to the destination line for the number of loop iterations specified. If the loop iteration count = 0, no jump will occur.

If the destination line number is greater than the line number of the loop command, the loop command acts as a GOTO and no looping will occur.

8.2.5 Ramp

The ramp command accepts three arguments: temperature, humidy, and time. This command causes the oven to change it's current temperature and humidity values to match the new values. The RAMP (rate of change) is determined by the time argument. For example, if the ramp command is executed when the oven is at 25 degrees C, and the target temperature is 50 degrees C, and if the time specified is 25 minutes, the oven will change at a rate of 1 degree C per minute until it reaches the target value.

8.2.6 Soak

Soaking is simply letting the oven achieve a stable temperature and humidity. The command accepts a time argument which is the time the oven will SOAK at the current temperature and humidity values.

8.3 Oven Synchronization

A main feature of the Orion test system software is it's ability to synchronize oven control with test scripts. The most important criteria in environmental test systems are to perform certain types of device operations at a specific temperature and/or humidity. The following explains how this is accomplished in the Orion test system.

8.3.1 Oven Slave Mode

The Orion test script can be made to slave to the oven. This means that the oven is free running and the port test script can synchronize to it. This is accomplished using the test script command WAIT FOR OVEN STEP.

The following figure shows an example of an oven script. Each oven instruction has an associated line number.

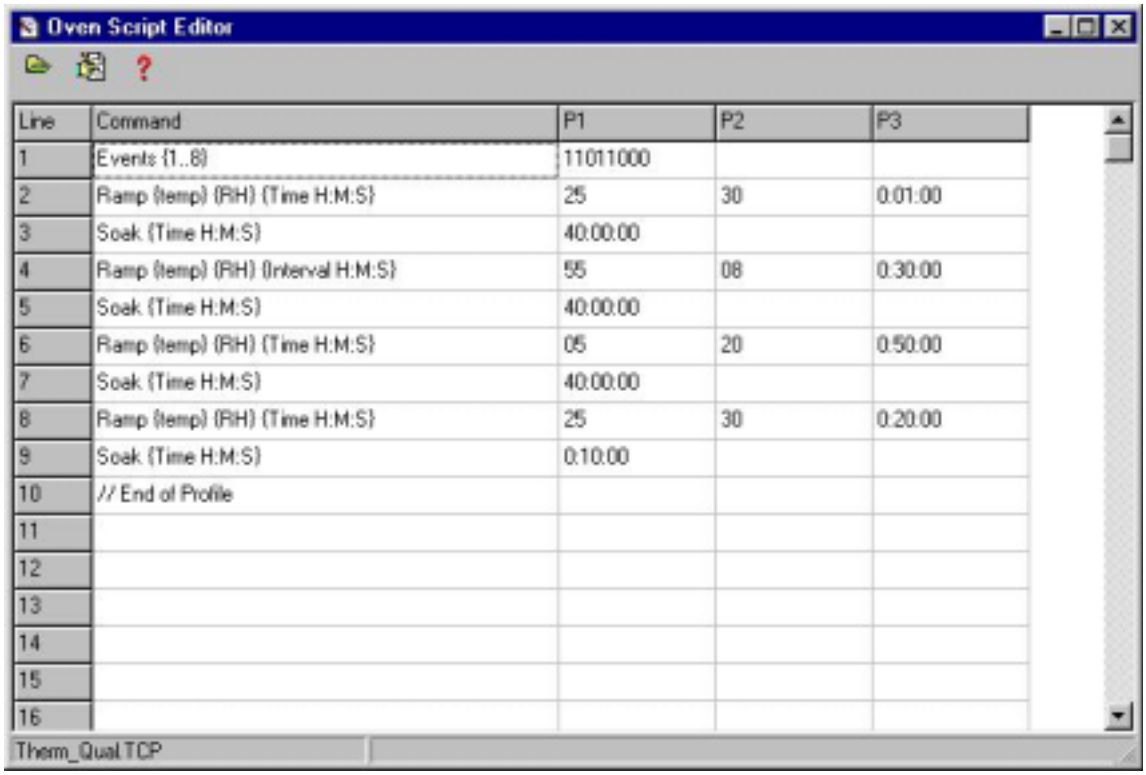


Figure 27- Oven Script Example

The test script command WAIT FOR OVEN STEP will wait for the oven script program to reach a specific line number or STEP before continuing execution. So the program example below will reach a point in its execution where it will wait for the oven program to reach line number 3:

```
001   RUN OVEN PROGRAM prog=BEG  line=1  // starts the default oven program
002 { Timed Loop sec=150
003 } End Timed Loop
004   WAIT FOR OVEN STEP=3  // wait for oven = 25 deg C
005   WAIT DELAY (x .1 sec)=100
006   --- more
```

Notice that the WAIT FOR OVEN STEP command assumes that an oven program is already running. If not, the test script will be held in an infinite wait condition because the oven script will never reach line 3.

Notice also that line 001 of the test script has the command RUN OVEN PROGRAM. This command is used here to start the oven script program set as the default (see section 6.2.1.3).

When using the oven slave mode, it is important to use appropriate ramp and soak times that prevent the test script program from getting *out of sync* with the oven program. For example:

- An oven script is programmed to cycle temperature and then soak for 20 minutes.
- A test script routine is designed to read and write during the 20-minute soak period then wait for the next ramp and soak before beginning another routine.

If the device takes longer than 20 minutes to perform its function, the oven will advance to its next operation while the device is still performing the R/W:

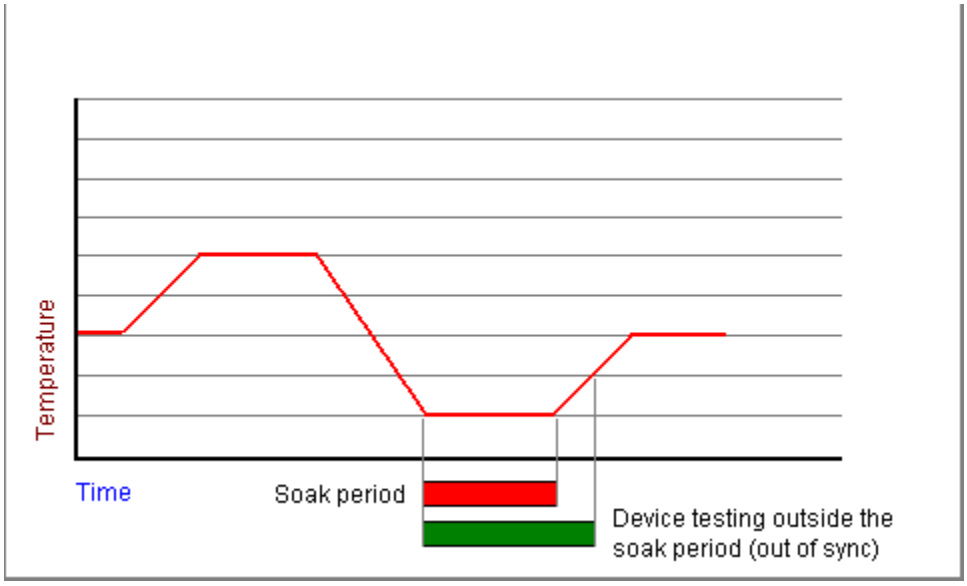


Figure 28 - Oven Slave Example

The solution to this problem is to set the SOAK period long enough to guarantee that the testing has been completed on the device.

8.3.2 Oven Master Mode

Another mode of oven synchronization is the so-called master mode. In this mode, the device test script can force the oven to advance to its next operation before the programmed period has expired. In the Oven Slave Mode the command RUN OVEN PROGRAM was used to start an oven script running. This command can also be used to advance a running oven script. The following device test script shows an example of this:

```
001   RUN OVEN PROGRAM prog=BEG  line=1  // starts the default oven program
002 { Timed Loop sec=150
003 } End Timed Loop
004   WAIT FOR OVEN STEP=3  // wait for oven = 25 deg C
005   WAIT DELAY (x .1 sec)=100
006   WRITE INCR FORWARD pass=1  // write while soaking at 25 deg C
007   RUN OVEN PROGRAM prog=INC  line=1  // force oven to advance
008   --- more
```

The above example will force the oven to advance from oven script line 3 to line 4. The result is a reduction in test time since the test script no longer waits for the oven to complete a conservative interval.

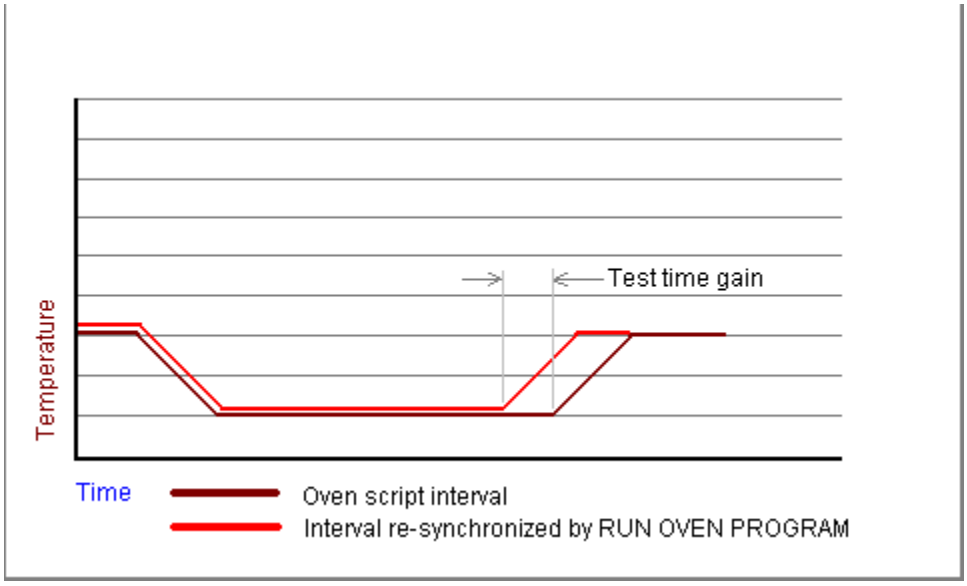


Figure 29 - Oven Master Example

When a *group* of ports are programmed to master the oven, the mastering can only be done when all ports in the *group* have reached the same point in their script execution. When an individual port reaches the RUN OVEN PROGRAM instruction, it will log into the Orion Host and wait for a *continue* message from the Host. At this time, a special icon is displayed on the Orion GUI to indicate the wait state: .

When all ports in the *group* have logged in, the oven script is advanced and a *continue* message is issued to all ports that have logged in. Some conditions will remove individual ports from the oven-mastering *group*:

- Port fails the test prior to reaching the RUN OVEN PROGRAM instruction.
- Port goes OFFLINE for some reason.

A watchdog timer is also provided to force test script and oven script continuation even though all ports in the group have not logged in. This feature prevents a lockout condition if a minority of ports are lagging behind the majority (see section 5.1.5).

Note also that both SLAVE and MASTER modes can be mixed within the same port test script.

9.0 Troubleshooting

This section provides some troubleshooting techniques designed to aid in the resolution of application and system problems.

9.1 Server will not communicate with clients

This indicates a basic problem with the Ethernet link. The following suggestions may help determine the problem:

1) Check the Host Setup menu and determine if the network I/O address matches the address setting on the NE2000 NIC card.
2) Check to see if the clients MAC address is correctly entered in the client properties table and that the client(s) are enabled.
3) Check to see that the Ethernet cable is connected properly from the server to the hub and that the Ethernet cable is of the proper type (straight through or crossover).

10.0 Technical Support

Online support and the latest releases of software are available on the Flexstar web site at http//www.flexstar.com.

Flexstar Technology
47323 Warm Springs Blvd.
Fremont, CA 94539
(510) 440-0170
(510) 440-0177 (fax)

Users can also receive technical support via Email at support@flexstar.com.

11.0 Glossary

ASCII - American Standard Code for Information Interchange. Universal character code.

ATA - Advanced Technology Attachment: This is the "official" name that ANSI group X3T10 uses for what the computer industry calls IDE (Integrated Drive Electronics) which is a low cost hard drive interface. This hardware interface is also the basis for ATAPI (see below).

ATAPI - AT Attachment Packet Interface – low cost CD-ROM interface – low cost IDE interface with a SCSI command packet protocol.

BFI – Bytes from Index. Format used in legacy systems but is not supported in Orion.

CCS – Common Command Set (SCSI).

CD – Compact Disk (CD-ROM).

CDB – Command Data Block.

CHS – Cylinder, Head, Sector physical format.

Client – Individual computers that perform device test. Clients are diskless computers that are networked to a central SERVER. Clients may have one or more test PORTS associated with it.

DUT – Device under test.

DVD – Digital Video Disk.

ESD – Electrostatic Discharge.

Ethernet – Standard LAN network hardware.

Fibre Channel – High speed serial interace. Used in high end systems and RAID applications.

GUI – Graphical User Interface.

HDD – Hard Disk Drive.

IDE - Integrated Drive Electronics. Low end HDD interface for home use.

LBA – Logical Block Addressing format.

Legacy – Flexstar FS25xx test systems (replaced by Orion).

Node – Used interchangeably with PORT.

PDF – Adobe Portable Document Format.

Port – Refers to a single device under test (DUT). There are one or more test ports per client.

Rack – Collection of clients that form a test system array.

Script – A text based programming language that executes within an interpreter. Scripts can easily be modified by the user to change the test system behavior.

SCSI - Small Computer System Interface. Interface for high performance systems.

Server – PC that performs control and data collection on the Orion system.

12.0 Appendix A - Image File Format

The image file (.IMG) is the primary database file of the Orion test system. The first 5200 bytes of the image file is a fixed structure, which is a combination of binary data and ASCII strings. The remainder of the file contains the error list. The list size is dynamic and each error in the list occupies 34 bytes of storage. Therefore, depending on the number of errors, the file can be very large or at a minimum, 5200 bytes.
A second file (.IM2) contains additional test information. Together, these two files are used by the report generator to build an ASCII report output file (.REP).

Primary Image File (.IMG)

Item	Data Type	Offset & Description	
IRACK	BYTE	0	Rack number
IPORT	BYTE	1	Port number
PROM_REV	WORD	2	Tester revision num (legacy)
DRV_TYPE	BYTE	4	Drive/port type
DRIVE_SN	string[20]	5	Device serial number
TEST_PGM	string[8]	25	Name of P25 script
START_TIME	ULONG	33	Start time stamp (legacy)
IDRV	BYTE	37	Drive #
TESTER F/W	string[4]	38	ASCII tester firmware string (legacy)
spare1	BYTE	42	(spare)
PORT_FAILC	BYTE	43	Port fail code
DefStructSize	BYTE	44	Sizeof defect structure
MAX_CYL_EXT	ULONG	45	Max Cyl Extended (>32 bits)
CREDENCE	BYTE	49	Hard error credence
TRACK_LEN	WORD	50	Track length
END_TIME	ULONG	52	End time stamp (legacy)
DRIVE_MN	string[10]	56	Device model number string
IOVEN_PROG	BYTE	66	Oven program number
IOVEN_STEP	BYTE	67	Oven step (line) number
IOVEN_TEMP	BYTE	68	Oven temperature
SECTOR_LEN	WORD	69	Sector/block length (in bytes)
MAX_CYL	WORD	71	Max cylinder #
MAX_HEAD	BYTE	73	Max head #
NUM_PASSES	WORD	74	Pass counter
SOFTREVNUM	WORD	76	Host s/w rev (legacy)
RESTARTFLG	BYTE	78	Restart flag (legacy)
FULL_TRK	WORD	79	Used by summary
LASTSTATUS	BYTE	81	Last status byte from the port
END_CYL	WORD	82	Last cyl on the device
END_HEAD	BYTE	84	Last head on the device
LINENUMBER	WORD	85	Current script line number
CMDNUMBER	BYTE	87	Cur cmd number (op code)
TEMP	WORD	88	Current temperature
CUR_LBA	ULONG	90	Current LBA

CUR_ZONE	BYTE	94	Current zone
NUMPASSES	WORD	95	MSW of loop counter
NUMECCBYTE	BYTE	97	Number of ECC bytes (IDE)
CTLR_TYPE	BYTE	98	Controller type (IDE)
spare3	BYTE	99	(spare)

*** Most recent test summary portion (offset 100)

IMG_UPNUM	WORD	100	Most recent update number
UPDATE_TIME	ULONG	102	Time stamp
NUM_SEEKS	ULONG	106	Seek counter
NUM_TRACKS	ULONG	110	
GROWNERRS	WORD	114	MAP ERRORS
NUM_SYNC	WORD	116	Id & data sync err counter
NUM_CMDTO	WORD	118	Number of command timeouts
NUM_CRC	WORD	120	ID/CRC errors
NUM_HARDHD	BYTE	122	# of hard errors per head (legacy)
BAD_HEAD	BYTE	123	Bad head, either soft or hard (legacy)
NUM_SOFTHD	BYTE	124	
NUMSYNC	BYTE	125	Sync Loss
NUM_HARDS	WORD	126	
NUM_SOFTS	WORD	128	# of soft errors
NUM_RNFS	WORD	130	# of RNF errors
NUM_SEEKER	WORD	132	# of seek errors
NUM_IDXFLT	WORD	134	# of index errors
NUM_SKCFLT	WORD	136	# of seek complete faults
NUM_WRTFLT	WORD	138	# of write faults
NUM_RDYFLT	WORD	140	# of not ready errors
NUM_SELFLT	WORD	142	# of selection errors
NUM_TK0FLT	WORD	144	# of track 0 faults
NUMTSTRFLT	WORD	146	# tester faults
N_ENTRIES	WORD	148	# defect list entries
MFG_MN	string[40]	150	Model number of device
FWREV	string[8]	190	Firmware rev of device
NUM_ABORTS	WORD	198	Number of aborted commands (PCAT)
SN_FILENAME	string[9]	200	Filename (s/n justified)
SECS_READ	ULONG	209	Msb of num_tracks
PROF_EVENT	string[4]	213	Event code (ascii) from power profile
CH0_VOLTS	WORD	217	Ch0 voltage in mV
CH1_VOLTS	WORD	219	Ch1 voltage in mV
PROF_RATE	WORD	221	Profile sample Rate in uS
MULTSECNUM	BYTE	223	# of multiple sectors supported (IDE)
CAPABILITIES	WORD	224	Capabilities word (IDE)
PIO_TIMING	BYTE	226	PIO timing (IDE)
DMA_TIMING	BYTE	227	DMA timing (IDE)
SPIN_RPM	WORD	228	Spindle RPM
INDEX_TIME	WORD	230	Index time in milliseconds
SEEK_TIMES	WORD[3]	232	1 tk, 1/3 stroke, max stroke timing (0.1 ms)
AVG_SEEK	WORD[3]	238	Average seek timing
RXFR_CRCS	WORD	244	IDE UDMA Read CRC errors
MAXCMDHIST	WORD	246	Max number of cmd history entries

EXTENDEFS	BYTEBOOL	248	Extended defect list support (T/F)
BYTEPERSEC	WORD	249	Bytes per sector (unformatted)
OAVG_SEEK	WORD[3]	251	Osc average seek timing
ATN_CODES	BYTE[4]	257	Esdi attention codes 4 bytes (legacy)
UNQ_CODES	BYTE[16]	261 -	Esdi vendor unique status 16 bytes (legacy)
POWER_STUF	BYTE[10]	277	Max 5I,time,nom 5 I,max 12I,time,nom 12 I
ZONE_NAME	string[13]	287 -	Name of zone table file - 13 bytes
SCSICMD_BLK	BYTE[13]	300 -	Byte 0 = Op code, 13 bytes
			1 = Sense key
			2 = Sense code
			3 = Sense qual
			4,5 = UEC
			6-9 = Error LBA
SPIN_UP	WORD	313	Spin up time in .1 sec incrs
STARTTIME	string[20]	315	Ascii start date & time
ENDTIME	string[20]	335	Ascii end date & time
DURATION	string[12]	355	Ascii test duration hhhh:mm:ss
AT_CONFIG	WORD	367	AT configuration word
DRQ_ERRORS	WORD	369	Max DRQ errs
IRQ_ERRORS	WORD	371	Max IRQ timeouts
BITS_READ	string[15]	373	Ascii bits read - 15 chars max
NUMSECSRD	INT64	388	8 byte long int sectors read
CUR_HEAD	BYTE	396	Current Head Number
UDMASUPRT	WORD	397	UDMA support word
spare5	BYTE	399	spare
SYSTEM_ID	string[16]	400	System ID (16 chars)
DEFC_DATA	BYTE[12]	416	Last data pattern used
NUM_VARS	WORD	428	# of Variables used
VAR_USED	BYTE[1194]	430	Variable buffer 1194 bytes
MIN_CYL	WORD	1624	Min cylinder
MIN_HEAD	BYTE	1626	Min head
NUM_HEADS	BYTE	1627	Number of heads
MIN_SECTOR	BYTE	1628	Min sector
CUR_SECTOR	BYTE	1629	Current sector
MAX_SECTOR	BYTE	1630	Max sector
NUMSECTORS	BYTE	1631	Number of sectors
MIN_LBA	ULONG	1632	Min LBA
MAX_LBA	ULONG	1636	Max LBA
PRIMARYERRS	WORD	1640	Number of primary errors
RECOVERRS	WORD	1642	Number of recov errors
NONRECERRS	WORD	1644	Number of non-recov errors
COMPR_ERRS	WORD	1646	Number of compare errors
NUMSKSOFT	WORD	1648	Number of soft seek errors
SOFTRATE	string[15]	1650	Ascii soft error rate string
HARDRATE	string[15]	1665	Ascii hard error rate string
SEEKRATE	string[15]	1680	Ascii seek error rate string
CutomerSpecific	BYTE[20]	1695	Customer specific area - do not use
REMOV_CYLS	WORD	1715	Removable cylinders
IDE_VENUNQ	BYTE[12]	1717	IDE Vendor unique str from ID drive data
SW_DMA	WORD	1729	Single word DMA (IDE)

MW_DMA	WORD	1731	Multiword DMA (IDE)
SCSIDEVICES	string[12]	1733	Device type (SCSI) ascii str 12 chars
ANSI_SPEC	BYTE	1745	ANSI spec (SCSI)
INITIATOR_ID	BYTE	1746	Initiator ID (SCSI)
TGT_ID_LUN	WORD	1747	SCSI target ID, & LUN
MFG_ID	string[8]	1749	Manufacturing ID (SCSI)
LBASPERTRK	WORD	1757	LBA's per track (SCSI)
SYNC_DATA	BYTE[6]	1759	SCSI SYNC data (req off, req per, neg per, neg off)
SEEKTRK1MIN	WORD	1765	1 track min seek timing
SEEKTHRDMN	WORD	1767	1/3 track min seek timing
SEEKFULLMIN	WORD	1769	Full track min seek timing
SEEKAVGMIN	WORD	1771	Average min seek timing
SEEKOAVMIN	WORD	1773	Oscillating average min seek timing
SEEKTRK1MX	WORD	1775	1 track max seek timing
SEEKTHDMX	WORD	1777	1/3 track max seek timing
SEEKFULLMX	WORD	1779	Full track max seek timing
SEEKAVGMX	WORD	1781	Average max seek timing
SEEKOAVMX	WORD	1783	Oscillating average max seek timing
CTLR_SIZE	WORD	1785	Calculated size of controller buffer (IDE)
ANSI_SPEC	string[16]	1787	Ansi Specification string (16 chars)
CYLS_TESTED	WORD	1803	Numbers of cylinders tested
MIN_MWDMA	WORD	1805	Minimum Multiword DMA timing (ns)
REC_MWDMA	WORD	1807	Recommended Multiword DMA timing (ns)
MIN_PIOWFLO	WORD	1809	Minimum PIO timing W/O flow control (ns)
MIN_PIOFLO	WORD	1811	Minimum PIO timing With flow control (ns)
CAPACITY	ULONG	1813	Drive capacity
CLIENT_VER	string[15]	1817	Client revision string
HOST_VER	string[15]	1832	Host Software version string
CUR_TEMP	WORD	1847	Current oven temperature
CUR_RH	WORD	1849	Current oven humidity
WXFR_CRCS	WORD	1851	IDE UDMA write CRC errors
BITS_WRITN	string[15]	1853	Ascii string – bits written
BLKS_WRITN	string[15]	1868	Ascii string – blocks (sectors) written
PROC_MSG	string[16]	1883	Ascii string – system process message
ERR_CODE	WORD	1899	Failure code
ERR_CLASS	WORD	1901	Failure class
CH0_CUR	WORD	1903	Ch0 current (ma)
CH1_CUR	WORD	1905	Ch1 current (ma)
CH2_VOLTS	WORD	1907	Ch2 voltage (mv)
CH2_CUR	WORD	1909	Ch2 current (ma)
CH3_VOLTS	WORD	1911	Ch3 voltage (mv)
CH3_CUR	WORD	1913	Ch3 current (ma)
4V_SUPPORT	BYTE	1915	1 = four channel power supported
CH2_MAXCUR	WORD	1916	Ch2 max current
CH2_MAXTM	BYTE	1918	Ch2 max time
CH2_NOMCUR	WORD	1919	Ch2 nominal current
CH3_MAXCUR	WORD	1921	Ch3 max current
CH3_MAXTM	BYTE	1923	Ch3 max time
CH3_NOMCUR	WORD	1924	Ch3 nominal current
spare6	BYTE[274]	1851	spares (274)

*** Test file chaining history: 8 bytes filename, 4 bytes time; up to 32 entries

CHAIN_FIFO	BYTE[2400]	2200	Chaining circ fifo (2400 bytes)
CHAINCOUNT	BYTE	4600	# chains done
CHAIN_PTR	BYTE	4601	Ptr to circ fifo
spare7	BYTE[16]	4602	Reserved (customer specific)
FAIL_STRING	string[50]	4618	Fault code string (50 chars)
CMD_STRING	string[50]	4668	Command string code (50 chars)
TOTALCHAIN	WORD	4718	Total chain count
FW_REVLONG	string[16]	4720	Long version of firmware rev
UserSetMinCyl	ULONG	4736	User set min cylinder
UserSetMaxCyl	ULONG	4740	User set max cylinder
UserSetMinHD	WORD	4744	User set min head
UserSetMaxHD	WORD	4746	User set max head
DEF_CTL_INF	ULONG	4748	*** See note 1 below
spare8	BYTE[273]	4752	spares (273)
CUST_SPEC	BYTE[16]	5025	Customer specific
Spare9	BYTE[159]	5041	spares
DEFECT_LIST	DYNAMIC	5200	Defect / error log

The defect list can be one of the following lengths:
34 – standard.
38 – 16 byte CDB support.
42 – 4V support.
46 – 16 byte CDB + 4V support.

The size of the defect list structure can be found at offset 44 decimal.

The following table describes the defect list structure as a function of its size:

Byte	Size = 34	Size = 38	Size = 42	Size = 46
0	Error type	Error type	Error type	Error type
1	LBA or CHS	LBA or CHS	LBA or CHS	LBA or CHS
5	Credence	Credence	Credence	Credence
6	SCSI ASC or IDE status	SCSI ASC or IDE status	SCSI ASC or IDE status	SCSI ASC or IDE status
7	Bit length	Bit length	Bit length	Bit length
8	Time (min)	Time (min)	Time (min)	Time (min)
10	Error code	Error code	Error code	Error code
11	SCSI sense key or IDE error reg	SCSI sense key or IDE error reg	SCSI sense key or IDE error reg	SCSI sense key or IDE error reg
12	Last cmd opcode	Last cmd opcode	Last cmd opcode	Last cmd opcode
13	Script line lsb	Script line lsb	Script line lsb	Script line lsb
14	Data written	Data written	Data written	Data written
15	Data read	Data read	Data read	Data read
16	SCSI ASQ	SCSI ASQ	SCSI ASQ	SCSI ASQ
17	Error offset	Error offset	Error offset	Error offset
19	Ch0 V	Ch0 V	Ch0 V	Ch0 V
21	Ch0 I	Ch0 I	Ch0 I	Ch0 I

23	Ch1 V	Ch1 V	Ch1 V	Ch1 V
25	Ch1 I	Ch1 I	Ch1 I	Ch1 I
27	Temperature	Temperature	Temperature	Temperature
29	Humidity	Humidity	Humidity	Humidity
30	Error offset msb	Error offset msb	Error offset msb	Error offset msb
31	Script line msb	Script line msb	Script line msb	Script line msb
34		Extended LBA (4 bytes)	Ch2 V	Ch2 V
36			Ch2 I	Ch2 I
38			Ch3 V	Ch3 V
40			Ch3 I	Ch3 I
42				Extended LBA (4 bytes)

Secondary Image File (.IM2)

Item	Data Type	Offset & Description	
NUM_CMDS	BYTE	0	Number of cmd history entries (# of structures)
SIZEOF_CHIST	BYTE	1	Size of command history structure
CHIST_FIFO		2	Start of array of structures
		8001	End of array of structures
Spares	BYTE[14]	8002	Spare area
CONFIG0	WORD	8016	Configuration Data Length
CONFIG	WORD	8018	General Configuration
BUFFER_TYPE	WORD	8020	Buffer Type
BUFFER_SIZE	WORD	8022	Buffer Size
CAPABILITIES	WORD	8024	Capabilities
PIO_TIMING	WORD	8026	PIO Cycle Timing
DMA_TIMING	WORD	8028	DMA Cycle Timing
SWDMA	WORD	8030	Single Word DMA mode
MWDMA	WORD	8032	Multi Word DMA mode
ENHAN_PIO	WORD	8034	Enhanced PIO mode
BLIND_PIO	WORD	8036	Blind PIO minimum Cycle
spare1	BYTE[10]	8038	(spare)
DEV_TYPE	BYTE	8048	Inquiry byte 0 (Device Type)
RMB	BYTE	8049	Inquiry byte 1 (RMB)
SPEC_VER	BYTE	8050	Inquiry byte 2 (Spec Versions)
RESP_DATA	BYTE	8051	Inquiry byte 3 (Responce Data)
spare2	BYTE[12]	8052	Spare
PG01_FLAG	BYTE	8064	Md Sns page 1 Flags byte
RD_RETRY	BYTE	8065	Md Sns page 1 Read Retry Count
INACTIVE_TMR	BYTE	8066	Md Sns page 0D Inactivity Time Mul
MSF_SEC_MIN	WORD	8067	Md Sns page 0D Sec per Min
MSF_FRM_SEC	WORD	8069	Md Sns page 0D Frame per Sec
PG0E_FLAG	BYTE	8071	Md Sns page 0E Flags byte(byte 2)
LBA_SEC_FMT	BYTE	8072	Md Sns page 0E LBA/Sec Format - This = 0 if bit 7 is 0
LBA_SEC_AUDIO	WORD	8073	Md Sns page 0E LBA/Sec playback - This = 0 if bit 7 is 0

CH_SEL0	BYTE	8075	Md Sns page 0E Port 0 Ch Sel
CH_VOL0	BYTE	8076	Md Sns page 0E Port 0 Volume
CH_SEL1	BYTE	8077	Md Sns page 0E Port 1 Ch Sel
CH_VOL1	BYTE	8078	Md Sns page 0E Port 1 Volume
CH_SEL2	BYTE	8079	Md Sns page 0E Port 2 Ch Sel
CH_VOL2	BYTE	8080	Md Sns page 0E Port 2 Volume
CH_SEL3	BYTE	8081	Md Sns page 0E Port 3 Ch Sel
CH_VOL3	BYTE	8082	Md Sns page 0E Port 3 Volume
CAP4	BYTE	8083	Md Sns page 2A Flags1(byte 4)
CAP5	BYTE	8084	Md Sns page 2A Flags2(byte 5)
CAP6	BYTE	8085	Md Sns page 2A Flags3(byte 6)
CAP7	BYTE	8086	Md Sns page 2A Flags4(byte 7)
MAX_SPD	WORD	8087	Md Sns page 2A Max Speed(Kbps)
VOL_LEVELS	WORD	8089	Md Sns page 2A Num Vol Levels Supp
BUF_SIZE	WORD	8091	Md Sns page 2A Buffer Size(Kbytes)
CUR_SPD	WORD	8093	Md Sns page 2A Current Speed(Kbps)
spare3	BYTE[17]	8095	Spare
NUM_ENTRIES	BYTE	8112	Number of TOC Track Descriptors
spare4	BYTE	8113	Spare
FIRST_TRACK	BYTE	8114	Rd TOC 1st Track Number
LAST_TRACK	BYTE	8115	Rd TOC Last Track Number Track Descriptors
START_TOC	BYTE[1000]	8116	array of TOC structures (100)
spare5	BYTE[4]	9116	Spare
GEN_AC_LEN	WORD	9120	Generic Access Structure Size
GEN_AC_ENT	WORD	9122	Number of Generic Access entries
START_GENAC	BYTE	9124	Room for 75 entries plus spare

The array of structures noted above (CHIST_FIFO) starting at offset 2, is the command history fifo which consists of up to 200 command history structures. The first structure is the most recent command sent to the device. The format of these structures is as follows:

Item	Size	Description
flag	byte	bit wise control flag: 0-3 = type, 4 = last FEATURE reg used, 5 = CDB packet data valid 6-7 = spare
len	byte	Length of remaining bytes
taskfile[8]	byte	0 = spare 1 = Feature reg 2 = Sector count reg 3 = Sector number reg 4 = Cyl low reg 5 = Cyl high reg 6 = Dev/Sel reg 7 = Command reg
cdb[16]	byte	CDB & packets (16 bytes)
altstatus	byte	ATA/ATAPI
status	byte	ATA/ATAPI
error	byte	ATA/ATAPI

int_rsn	byte	ATAPI
key	byte	SCSI
asc	byte	SCSI
asq	byte	SCSI
cdb_len	byte	SCSI (length of CDB)
cmdinfo[6]	byte	future use (6 bytes)

13.0 Appendix B – Interpreting the Test Report

At certain times, the Orion test system will generate an ASCII test report. The report is saved as a file in the directory selected for ***Report Files*** (see section 5.1.3). The report files are usually named using the serial number of the device being tested and with the file extension ".REP". If the serial number is not available for some reason the file is named using the port physical position and has the format RxxPyyD0.REP where ***xx*** is the RACK number and ***yy*** is the PORT number.

Report files are generated under the following conditions:

- Test conclusion (PASS or FAIL).
- Manual report generated using the Generate Report button on the Orion toolbar.
- Image file updates – this can be disabled via the setup menu.

Test reports can optionally be sent to the system printer by selecting this option in the setup menu.

The test report is generated by the built-in report generator that uses a scripting language to determine content and format (see section 7.0). The following is a description of a typical report based on the standard report script (*.ORP) files that are delivered with the system.

```
Orion Report Generator  V 1.5.0.1    [Report generator version #]
11/06/2000 11:23:13    [Report date & time]

                ORION TEST REPORT FOR RACK  00 Port 00 Drive 00    [Port physical location]

                HOST VER:  1.4.3.10   CLIENT VER:  01.01.09Beta15    [Host & client versions]

System ID: abc    [System ID field – see section 5.1.1]
Process Message: def    [Comment string – see section 5.1.1]

                \\\ DRIVE CONFIGURATION ///    [Drive reported configuration info]

Model#:          SAMSUNG VG34805A    (4.80GB)
Serial#:         V4_C
Configuration:   045A
Fixed Cyls:      16544           Remov Cyls:    00000
Heads:           09              Sectors/Trk:   063
UnFmt byte/Tk:   32256           Bytes/Sector:  512
BufferType:      03              BufSize:       00000
ECC Bytes:       00              Drv F/W Rev:   j3090
Mult Secs/Blk:   016             Capacity:      008F2260
Capabilities:    0F00            Ven Unique:    0000
PIO Timing:      03              DMA Timing:    07
Sngl Wd DMA:     0007            MultiWrd DMA:  0007

Line:  15              Current Cmd: (154) TEST SEEK TIMING    [Last test script line#]
[Command history fifo – up to 255 low level commands]
```

```
ATA-Cmd     Cyl     DvHd    Snum    Scnt    DvCt    Alt     Stat    Err     Ftr
70 SEEK      0      A0      1       0       00      50      50      00
70 SEEK      100    A0      1       0       00      50      50      00
70 SEEK      0      A0      1       0       00      50      50      00
70 SEEK      100    A0      1       0       00      50      50      00
70 SEEK      0      A0      1       0       00      50      50      00
70 SEEK      100    A0      1       0       00      50      50      00
70 SEEK      0      A0      1       0       00      50      50      00
70 SEEK      100    A0      1       0       00      50      50      00
```

[Drive test info – M/N, S/N, test name, start date, chain history, etc.]

```
------------------------- DRIVE TEST INFORMATION -----------------------------
User MN     S/N                Type   Test_Pgm Start_Date/Time  Duration(H:M:S)
           V4_C                IDE      ide_2    11/06/2000 11:22:42  00:00:29
                                        startup  11/06/2000 11:24:12  00:01:43
```

[Error counters show totals for head/media related errors]

```
------------------------------ ERROR COUNTERS --------------------------------
Hard   Soft   Primry Grown  Recov  NonRec Compar SeekH  SeekS  Sync   RNF/AMNF
0      0      0      0      0      0      0      0      0      0      0

ID/DATA CRC    Rd Xfer CRC     Wr Xfer CRC
0                    0               0
```

[Error rate calculations- bits transferred / errors]

```
------------------------- ERROR RATE CALULATIONS -----------------------------
Soft_Err_Rate  Hard_Err_rate    Seek_Err_Rate
0.00000E+000   0.00000E+000     0.00000E+000
```

[error counters for electro-mechanical faults]

```
------------------------------ DRIVE FAULT ERRORS ----------------------------
Index  Ready  Trk_0  Write  Select SkCmp  CmdTm0 Abortd
0      0      0      0      0      0      0      0
```

[Test counters keep track of bits transferred, etc.]

```
------------------------------- TEST COUNTERS --------------------------------
Loops    Seeks     Blocks_written  Bits_read       Bits_write
0        3937      0.00000E+000    2.34570E+008    0.00000E+000
```

[Seek timing results table]

```
--------------------------- TIMING TEST RESULTS ------------------------------
        RPM     INDEX   1tk_SK  1/3_SK  Max_SK  Avg_SK  Osc_SK  Spin_Up
                 (ms)    (ms)    (ms)    (ms)    (ms)    (ms)    (sec)
Min                     1.4     2.3     3.1     0.8     1.7
Max                     2.4     3.9     3.9     3.9     3.9
Avg     0       0       1.6     2.6     3.2     2.5     2.8     0.8
```

[Power module results – last state of the margined power outputs]

```
--------------------------- POWER MODULE RESULTS -----------------------------
5V Max Cur  5V Cur      12V Max Cur     12V Cur     5V Voltage      12V Voltage
0.584        0.39         1.195           0.446          4.977           11.954
```

[Oven temperature & humidity (last state)]

```
------------------------------- OVEN VALUES ----------------------------------
Temperature    Humidity
0000           0000
```

[Variables used within the test script]

```
------------------------------ VARIABLES USED --------------------------------
VC = 136F0240          VD = 2EAF00DB          TP = 00000000
SP = 00000008
```

[Defect list displays details of any errors detected]

```
------------------------------- Defect Map -----------------------------------
DATA=4E654E654E654E654E654E65
LBA <HCS>      Offset BL Rdat Wdat Cred Status Err Cmd  Line Type    Time(H:M)
```

[Final results – PASS or FAIL with error type description]

```
*** FINAL TEST RESULTS:  Test passed
```

```
[Performance results are appended to the end of the report]
================================================================================
>>> Performance data for file: V4_C.PRF
================================================================================

----------------------
DEMO                 |   11/06/2000  11:23:06
----------------------
Num Cmds: 303
Blocks Xfrd:      57267
Block Size: 512
Total Time(sec):  11.4489
Time/Block(us):   199.9
Rate KBytes/Sec:  2561.0
```

Figure 30 - Interpreting the Test Report

Section Details

[Error counters show totals for head/media related errors]

```
------------------------------ ERROR COUNTERS ---------------------------------
Hard   Soft   Primry Grown  Recov  NonRec Compar SeekH  SeekS  Sync   RNF/AMNF
0      0      0      0      0      0      0      0      0      0      0

ID/DATA CRC    Rd Xfer CRC     Wr Xfer CRC
0                    0               0
```

[Hard]
The hard error counter represents the total hard errors that are detected by the test system. Hard errors are repeatable errors that occur in the same physical location (CHS/LBA). Most errors on intelligent interfaces are treated automatically as HARD errors.

[Soft]
The soft error counter represents the total number of soft errors detected by the test system. Soft errors are non-repeatable errors. These types of errors are mainly detectable on older primitive interfaces (ST506, ESDI, etc.).

[Primry]
Primary errors are drive media related errors that are typically read from the drives P list. These are errors that the drive has already identified and mapped.

[Grown]
Grown errors are errors that are detected by the device or test system and have been added since the creation of the PRIMARY list.

[Recov]
Recoverable errors are drive reported errors that can be recovered by the use of ECC or other correction methods.

[NonRec]
Non-recoverable errors are drive reported errors that cannot be corrected by the drive. These errors are

always considered to be HARD errors.

[Compar]
Compare errors occur when data received from the drive (in the read buffer) does not match the contents of the write buffer (data written to the drive).

[SeekH]
The hard seek error counter is maintained for legacy compatibility but is not supported in Orion.

[SeekS]
The soft seek error counter is maintained for legacy compatibility but is not supported in Orion.

[Sync]
The Sync error counter is used to show the total number of SCSI Sync loss errors.

[RNF/AMNF]
The Record Not Found / Address Mark Not Found error counter records totals for both types of errors. A typical reason for these errors is trying to operate outside of the valid range of CYL, HEAD, or SECTOR.

[ID/Data CRC]
The ID/Data CRC counter is used to show the total number of Data or ID errors reported by the drive during testing.

[Rd Xfer CRC]
This counter shows the total number of UDMA CRC errors that occurred during a READ operation.

[Wr Xfer CRC]
This counter shows the total number of UDMA CRC errors that occurred during a WRITE operation.

[Error rate calculations- bits transferred / errors]

```
------------------------- ERROR RATE CALULATIONS -----------------------------
Soft_Err_Rate  Hard_Err_rate    Seek_Err_Rate
0.00000E+000   0.00000E+000     0.00000E+000
```

[Soft Error Rate]
This expresses the SOFT error bit error rate. These are the number of errors per bit transferred.

[Hard Error Rate]
This expresses the HARD error bit error rate. These are the number of errors per bit transferred.

[Seek Error Rate]
This expresses the SEEK error bit error rate. These are the number of errors per seek. Note that this counter is not supported on Orion and is only present for legacy compatibility.

[error counters for electro-mechanical faults]

```
----------------------------- DRIVE FAULT ERRORS -----------------------------
Index  Ready  Trk_0  Write  Select SkCmp  CmdTm0 Abortd
0      0      0      0      0      0      0      0
```

[Index]
This counter contains the number of index faults (ST506 only).

[Ready]
The ready counter shows the total number of NOT READY errors detected by the test system.

[Trk 0]
The Trk 0 counter shows the total number of RECALIBRATE errors (ST506 only).

[Write]
The Write counter shows the total number of WRITE FAULT errors reported by the drive (ST506).

[Select]
The Select counter shows the total number of select errors (failure to select the drive).

[SkCmp]
The SkCmp counter shows the total number of SEEK COMPLETE errors. These errors are generally discovered after a seek has been commanded and the drive does not assert seek complete status.

[CmdTmO]
The CmdTmO counter shows the total number of command timeout errors. A command timeout error occurs when a command of any kind has been issued but the drive remains busy or not seek complete.

[Abortd]
The Abortd counter shows the total number of COMMAND ABORTED errors. This error applies to IDE devices only and usually indicates that an invalid command or task file register value was sent to the drive.

[Test counters keep track of bits transferred, etc.]

```
------------------------------ TEST COUNTERS ---------------------------------
Loops    Seeks     Blocks_written  Bits_read       Bits_write
0        3937      0.00000E+000    2.34570E+008    0.00000E+000
```

[Loops]
This counter shows the total number of LOOPS performed inside a script START LOOP … END LOOP construct.

[Seeks]
Shows the total number of seeks performed by the drive.

[Blocks Written]
Displays the total number of BLOCKS (sectors) that have been written.

[Bits Read]
Displays the total number of BITS transferred from the drive.

[Bits Write]
Displays the total number of BITS transferred to the drive.

[Seek timing results table]

```
---------------------------- TIMING TEST RESULTS -----------------------------
        RPM     INDEX   1tk_SK  1/3_SK  Max_SK  Avg_SK  Osc_SK  Spin_Up
                 (ms)    (ms)    (ms)    (ms)    (ms)    (ms)    (sec)
Min                     1.4     2.3     3.1     0.8     1.7
Max                     2.4     3.9     3.9     3.9     3.9
Avg     0       0       1.6     2.6     3.2     2.5     2.8     0.8
```

[RPM]
Displays the disk rotational rate in RPM. ST506 / ESDI only.

[Index]
Displays the time from one index pulse to the next in milliseconds. ST506 only.

[1Tk SK]
Displays the results of the one track seek timing in the SEEK TIMING TEST macro.

[1/3 SK]
Displays the results of the 1/3 track seek timing in the SEEK TIMING TEST macro.

[Max SK]
Displays the results of the max track seek timing in the SEEK TIMING TEST macro.

[Avg SK]
Displays the results of the AVG SEEK TIMING TEST macro.

[Osc SK]
Displays the results of the OSC SEEK TIMING TEST macro.

[Spin Up]
Displays the results of the TEST SPINUP macro (time from power on to drive ready).

[Power module results – last state of the margined power outputs]

```
---------------------------- POWER MODULE RESULTS -----------------------------
5V Max Cur  5V Cur      12V Max Cur     12V Cur     5V Voltage      12V Voltage
0.584        0.39         1.195           0.446          4.977           11.954
```

[5V Max Cur]
Displays the maximum amount of 5V current measured during the TEST SPINUP macro.

[5V Cur]
Displays the average amount of 5V current measured during the TEST SPINUP macro.

[12V Max Cur]
Displays the maximum amount of 12V current measured during the TEST SPINUP macro.

[12V Cur]
Displays the average amount of 12V current measured during the TEST SPINUP macro.

[5V Voltage]
Displays the voltage on the 5V channel measured during the TEST SPINUP macro.

[12V Voltage]

Displays the voltage on the 12V channel measured during the TEST SPINUP macro.

[Oven temperature & humidity (last state)]

```
------------------------------ OVEN VALUES ----------------------------------
Temperature    Humidity
0000           0000
```

[Temperature]
Displays the last measured temperature before test end.

[Humidity]
Displays the last measured humidity before test end.

[Variables used within the test script]

```
---------------------------- VARIABLES USED --------------------------------
VC = 136F0240          VD = 2EAF00DB          TP = 00000000
SP = 00000008
```

All variables that have been used in the test script are shown here.

[Defect list displays details of any errors detected]

```
------------------------------ Defect Map ------------------------------------
DATA=4E654E654E654E654E654E65
LBA <HCS>      Offset BL Rdat Wdat Cred Status Err Cmd  Line Type    Time(H:M)
```

[LBA <HCS>]
Logical Block Address or Head/Cylinder/Sector address is shown for the error logged.

[Offset]
Shows the offset in bytes where the error occurred. This only applies to data compare errors.

[BL]
Bit length of an error (ST506 only).

[Rdat]
Data read value (bad data) – data compare error only.

[Wdat]
Data wrote value (expected data) – data compare error only.

[Cred]
Hard/Soft error credence assigned in the ABORT LIMITS section.

[Status]
Displays the value of the status register contents when the error occurred.

[Err]
Displays the value of the error register when the error occurred.

[Cmd]
Opcode of the drive command that was associated with the error.

[Line]
Script line where the error was detected.

[Type]
Error type (DATA COMPARE, RNF, etc.).

[Time]
The time when the error occurred relative to the start time of the script.

Note that for each defect, a second line is added which has the following format:

```
→ch0_V 4.99 ch0_I 0.825 ch1_V 11.98 ch1_I 1.530 Temp>0024 Humid>0035
```

This second line indicates the voltage and current at the time of the error detection plus oven temperature and humidity.

14.0 Appendix C – Client Console Function

The client console function creates video and keyboard re-direction for a selected client. This allows the client to communicate via the server without having to connect a monitor or keyboard directly to the client motherboard.

This function is enabled via the Orion Host setup menu under the preferences tab (see section 5.1.6). The entry in the ***Client Console*** edit box is usually a Flexstar network re-direction utility called TNAW.EXE. This utility must exist in the Orion application directory (usually C:\Program Files\Flexstar\Orion). In order to enable the function, the ***Enabled*** checkbox must be checked.

In addition, a utility provided by Novell, NWSIPX32.DLL must be present in the WINDOWS\SYSTEM or WINDOWS\SYSTEM32 directory.

Note: The console window must be closed using either the F10 key or ALT-X. Closing the window using the normal method (i.e. clicking the close window button on the title bar) will result in the window closing but the application will remain in memory.

Warning: This function is included for debug purposes only and may not be stable in certain OS configurations. Flexstar does not recommend or support the use of this function.

15.0 Appendix D - Data Compare Description

The Orion tester utilizes software data compare to validate data integrity during WRITE / READ testing.

This is done by first writing a known pattern to the Device Under Test (DUT). The desired pattern is built in the WRITE BUFFER before the WRITE command is issued to the device. As the WRITE command is running, the data from the WRITE BUFFER will be sent to the device to write into its media.

The data is read back either in a sequential or random fashion using READ commands. Before the READ command is issued to the device, the WRITE BUFFER is rebuilt (if necessary) with the data that was written to the device. During the READ command, the data is received from the device and is read into the READ BUFFER of the tester. The two buffers are compared by software to verify that the data read back is the same as the written data.

There are a several pre-defined data patterns supported by the tester. The following describes the differences and how it affects the data build/compare operation.

Data Pattern Commands

Currently, there are six commands that build data patterns in the write buffer. They are:

1-8 Byte Pattern (builds repeating one to eight byte user defined data patterns)

2 Byte Pattern (builds a 2 byte user defined repeating data pattern)

12 Byte Pattern (builds a 12 byte user defined repeating data pattern)

Set Data Mode (supports several fixed patterns – see below and section 6.2.1.4)

Set Random Pattern (random data repeated every sector)

Set Sector Overlay (32 bit LBA or CHS embedded into each sector)

The *1-8 Byte, 2 Byte, and 12 Byte pattern* commands are very similar in that the buffer data pattern is built around the sequence defined within the 2, 8, or 12 byte sequences. The only difference is in the byte length and thus the pattern-repeat frequency in the buffer. 1-8 Byte pattern allows you to specify the length of the words between 1 to 8 bytes.

***Set Data Mode* support the following modes**

0 - User defined (set by the SET DATA PATTERN commands). This is the default condition.

1 - Random data (no seed).

2 - Random data (LBA seed).

3 - Incrementing data pattern.

4 - Decrement data pattern.

5 - Walking 1's data pattern (0x80, 0x40, 0x20, etc...).

6 - Walking 0's data pattern (0x7F, 0xBF, 0xDF, etc...).

The ***Set Random Pattern*** command is the same as *Set Data Mode* 1. Mode 1 builds a pseudo random sequence for 1 logical block which is then repeated throughout the buffer. Thus the data is pseudo random but repeats every logical block of data. Mode 2 is different in that each block of data uses the Logical Block Address (LBA) as part of the sequence to make each block unique. This makes the data pattern random and unique for each block.

The ***Set Sector Overlay*** function enables the option of overlaying the first or the last (or both) 4 bytes of the logical block data with the value of that LBA. This command has the following options:

0 - Sector overlay at the START and END of the block.

1 - Sector overlay at the START of the block.

2 - Sector overlay at the END of the block.

3 - Disable sector overlay (default condition).

Performance Overhead

There is always some overhead associated with software generated data pattern construction and software data compare. Due to the unpredictable nature of the combination of script sequences that can be created, the reference buffer (write buffer) is rebuilt per the pattern that is active before interface commands are sent to the DUT.

Due to the requirement to rebuild the buffer on every command (WRITE to build the reference buffer and READ to rebuild the reference for data compare), Data Mode 2 and Set Sector Overlay Mode 0 have the highest impact on overhead time especially if BOTH are used.

Data Mode 1 without Sector Overlay is one of the lowest overhead penalties because the buffer does not need to be rebuild as often.

Data Compare

The current Orion architecture utilizes software data compare. In this scheme, the WRITE BUFFER is used as the reference data because it was used to write the DUT with. When the DUT is read (device sends data to the testers READ BUFFER), the WRITE BUFFER and the READ BUFFER are compared against each other for the length of data bytes transferred for the current read command. If the data does not compare, then the offset into the buffer where the mis-compare occurred is logged as well as the data byte from the WRITE BUFFER (WDAT), and the READ BUFFER (RDAT).

Disabling Data Compare

In some cases, the command may return data but is not related to data written to the DUT (ie. Mode Sense, Inquiry, etc.). In these cases, the script must instruct the logic not to perform data compare. This is done by using the command *DATA COMPARE* (0=on 1=off) =1. This may also be needed in cases where the test must start with READS (before writing reference data). In these cases, the test is looking

for other READ related issues and the data integrity is not the interest.

By default the tester starts each test with Data Compare disabled.

Data Build/Rebuild Sequence

By default the data in the WRITE BUFFER is filled with the pattern "0x4E65" at the beginning of a test. The various data pattern commands can be issued to change this before WRITE/READ commands are issued to the DUT.

The data build/rebuild sequences may differ from data patterns selected. This section will describe the sequences of the various possible modes.

- **1-8 Byte Pattern**

  In this command, the user can select between 1 to 8 bytes as a length for the pattern. In this command, the user must specify the pattern as the second parameter. When this command is executed, it will take the user-specified pattern and fill the WRITE BUFFER with the pattern. Unless Sector Overlay is enabled, the WRITE BUFFER will not be rebuilt during WRITE or READ commands after this command has been executed.

- **Byte Pattern**

  In this command, the user must specify 2 bytes in the parameter to make up the pattern. When this command is executed, it will take the user-specified pattern and fill the WRITE BUFFER with the pattern. Unless Sector Overlay is enabled, the WRITE BUFFER will not be rebuilt during WRITE or READ commands after this command has been executed.

- **Byte Pattern**

  In this command, the user must specify 12 bytes in the parameter to make up the pattern. When this command is executed, it will take the user-specified pattern and fill the WRITE BUFFER with the pattern. Unless Sector Overlay is enabled, the WRITE BUFFER will not be rebuilt during WRITE or READ commands after this command has been executed.

Set Data Mode

- **Mode 0**

  Mode 0 turns off data pattern generation. This mode can be used once a data pattern was selected but when the user wants to download or build a specific data in the WRITE BUFFER to send to the DUT. This mode will keep the tester from generating some data pattern that would overwrite the user specified data. This is the default mode and if the user never selected a data pattern at the beginning of the test, then after the default "0x4E65" pattern built before the test starts, it will not regenerate the data pattern.

- **Mode 1**

  Mode 1 creates a pseudo random data pattern that is built by filling a logical block (sector) in the first section of the WRITE BUFFER. This is then block copied to all the rest of the WRITE BUFFER. Thus the data in each logical block would be exactly the same, but the content of the

block is made up of a pseudo random sequence. Without Sector Overlay enabled, this mode would not generate sector unique data.

- **Mode 2**

Mode 2 makes each logical block unique. For this mode, the data pattern is not built in the WRITE BUFFER when the Set Data Mode command is executed but is built just before the command (WRITE / READ) is issued to the DUT. In this mode, not only is the data pattern pseudo random, but each logical blocks is different from the rest. This will build the pattern only up to the length of the data transfer for that command.

- **Mode 3 & Mode 4**

Modes 3 and 4 will create incrementing or a decrement pattern respectively. The data pattern will start from 0x00 and increment until it reaches 0xFF. Then the pattern repeats again from 0x00 until the whole WRITE BUFFER has been built with the pattern.

- **Mode 5**

Mode 5 creates a walking 1's data pattern. In this mode the sequence would be 0x00, 0x80, 0x40, 0x20, etc ... such that each subsequent byte will have the next lower bit turned on and the rest off. When this pattern reaches 0x00 it then starts over with 0x80. This mode will fill the entire WRITE BUFFER with the pattern.

- **Mode 6**

Mode 6 creates a walking 0's data pattern. In this mode the sequence would be 0xFF, 0x7F, 0xBF, 0xDF, etc ... such that each subsequent byte will have the next lower bit turned off and the rest on. When this pattern reaches 0xFE and then starts over with 0xFF. This mode will also fill the entire WRITE BUFFER with the pattern.

Set Random Pattern

This command is the same as Set Data Mode 1.

Sector Overlay

If Sector Overlay is enabled, the overlay logic will generate an overlay double word into the beginning and/or the end of each sector (block) throughout the entire WRITE BUFFER before the command is issued. This done for WRITE commands and also for READ commands with DATA COMPARE enabled.

Software Data Compare

Software data compare is done if the Data Compare Enable command has been executed with the enable option. Only commands that do a DATA IN phase will have software data compare check after the command is complete.

Only the portion of data that was transferred will be compared from the READ BUFFER against the WRITE BUFFER. This amount is calculated based on the transfer length expected from the command

and how much was actually transferred by the Host Controller (HBA) according to the low level drivers.

The following script commands will execute software compare if enabled:

READ CURRENT TRACK

READ CURRENT CYL

READ INCR FORWARD

READ INCR REVERSE

READ OSCILLATE

READ RANDOM TRACK

READ RANDOM CYLIN

READ SECTOR

ATA VEN_UNQ commands with 'I' terminator.

Both SCSI and ATAPI VEN UNQ commands with 'I' terminator.

The software compare is only executed if the command sent to the DUT and completes without error. If the command was aborted or completed with some kind of error condition, the software compare will not be executed.

16.0 Appendix E – Command Timeout Description

Command timeouts tend to be one of the more common errors encountered in disk drive testing. This type of error occurs when a command is issued but no handshake or response is detected from the device within a given timeout period. The error type reported under these conditions will be COMMAND COMPLETE TIMEOUT. Also, there are two types of default status codes used to further identify the nature of the timeout.

In the Orion test system (when a command is issued) the actual status returned from a device cannot be predicted. Therefore, a placeholder value is substituted. This placeholder initially is an arbitrary value of A6 hex (SCSI only). This value will be replaced with an A5 hex following the selection phase. Therefore a status value of A6 indicates that a SCSI selection phase was never completed (selection timeout).

A value of A5 hex indicates that the selection phase completed successfully but the status phase following device command completion was never detected. The status phase is a different protocol for different interface types but is always the final phase following issuance of a command.

Other status codes are derived from the device and indicate that the status phase was completed. This code may also indicate some kind of device-detected error.

17.0 Appendix F – On Error Gosub Description

The ON ERROR GOSUB feature allows the user to create a custom error handler within the Orion script environment. When enabled, the script will jump to the function defined in the ON ERROR GOSUB instruction whenever an error occurs. The script syntax is:

ON ERROR GOSUB label=aaaaaa opt=xxxx

Where *aaaaaa* is the label of the new error handler and *xxxx* is the option selection.

Options are defined as follows:

xxxx = 0000: Perform operations inside the custom error handler and return to the next script instruction (ignore Orion default error handler).

xxxx = 0001: Perform operations inside the custom error handler and then call the Orion default error handler to augment error reporting.

*** Note that chaining to another script program is not allowed within the custom error handler.

18.0 Appendix G – Soft/Hard Error Description

A **SOFT** error by Flexstar definition is an error (usually media related) that does not occur repeatedly during error scans. **SOFT** errors can be caused by electrical noise, particulates passing randomly under the R/W head, marginal media defects that are not usually detectable, etc. Conversely, **HARD** errors are errors that occur repeatedly in the same physical location.

Every time an error occurs in the same physical location (CHS or LBA), the errors **HIT** counter is incremented. A threshold value known as **CREDENCE** (located in the ABORT LIMITS section) is used to determine if a particular error is soft or hard. If an error and its **HIT** count rises to be >= **CREDENCE**, then the error is converted from **SOFT** to **HARD**. If **CREDENCE** is set to 0 or 1, all errors automatically become **HARD** errors even though the **HIT** count is only = 1.

Although **SOFT** errors were an important issue on early disk drive interfaces such as ST-506, ESDI, etc., intelligent drive interfaces such as IDE and SCSI hide soft errors from detection through the interface. There are several references in this document to **SOFT ERRORS** but soft errors are not normally detectable in the newer intelligent interfaces and these references are maintained for legacy compatibility. Note also that HARD and SOFT seek errors are not supported in Orion and are only referenced for legacy compatibility.

19.0 Appendix H – Error Rate Calculations

Actual Bit Error Rates (BER) are calculated from the amount of data bits transferred divided into the number of errors that occurred. The number of BITS transferred is derived from the number of BLOCKS transferred multiplied by the block length (bytes per block) times 8 bits per byte. This number is then divided into the number of errors to obtain ERRORS PER BIT (also know as BER). The calculation is therefore expressed as follows:

BER = ERROR COUNT / (BLOCKS XFER * BYTES/BLOCK * 8)

Example: If there had been 100000 blocks transferred with 512 bytes per block and 2 errors had occurred within this transfer the BER would be:

BER = 2 / (100000 * 512 * 8) = <4.88 E-09>

20.0 Appendix I – Large File Download Example

This example demonstrates the ability of the Flexstar Orion test system to transfer large data files from the Orion Host to the device under test though a SCSI write buffer command.

This example assumes the following:

- A SCSI device that supports microcode downloading though the SCSI-2 Write Buffer command.
- Flexstar Orion Client software running on the tester port.
- Flexstar Orion Host software running on the server PC.
- The appropriate microcode file for the SCSI device copied into the O25Host directory in the server PC (this is obtained from the device vendor).
- SCSIDLD.P25 and SCSIDLD.PTX script loading into the appropriate directory of the server PC.Script

This script utilizes a feature in the client software, which allows the tester to combine the Write and Read track buffers into one large contiguous buffer. This way the tester affords up to a Megabyte buffer for the use of downloading vendor unique data to the device. Setting bit 3 of the “DB” variable to 1 enables this feature. This is accomplished by using the "SET VARIABLE" command on line 23 of the example script. This feature also allows the client software to attempt to transfer more bytes than either buffer would normally allow.
NOTE: This will only work with Vendor Unique or CCS commands and not with MACRO type commands.

```
001 * ------------------------------
002 * SCSI MICROCODE DOWNLOAD
003 * DEMO FOR ORION
004 * ------------------------------
005   SET VARIABLE v=SA (x)n=00000014
006   SET VARIABLE v=SD (x)n=00000014
007   SET Ch1 Volts=12.000 Reg#=0
008   SET Ch0 Volts=5.000  Reg#=0
009   SET CMD TIMEOUT time(sec) =240
010   SET DRIVE #/TARGET ID =16
011   TEST SPIN-UP
012   CONFIGURE PORT
013   SET SCSI mode=2
014   SYNC OP MODE offset=15  per=10
015 * DISABLE DATA COMPARE
016   DATA COMPARES (0=on 1=off) =1
017 * SET THE FILENAME OF THE
018 * MICROCODE FILE  -
019 * FILE MUST BE IN THE HOST FILE
020 * DIRECTORY
021   SET FILENAME =GSPMT03C.BIN
022 * ENABLE LONGXFER OPTION
023   SET VARIABLE v=DB (x)n=00000008
024   DNLD FILE
025   MOVE VAR s=DL TO VAR d=X0
```

```
026   WAIT DELAY (x .1 sec)=20
027   VEN UNQ 3B,05,00,00,00,00,X3,X2,X1,00,O
028   WAIT DELAY (x .1 sec)=50
029   RESET DRIVE
030   WAIT DELAY (x .1 sec)=50
031 * UPDATE REV LEVEL
032   CONFIGURE PORT
033   WAIT DELAY (x .1 sec)=10
```

Once the file has been downloaded into the client buffer with the "DNLD FILE" command in line 24, the variable "DL" will hold the size of the file downloaded into the buffer in bytes.

The following steps are performed to issue the SCSI-2 Write Buffer command with the 'O' terminator to direct the client to send the data from it's write buffer (in this case, write and read buffer). Note that the "X0" variable holds the value from "DL" variable, and that the Write Buffer command uses X3, X2, X1 in place of the transfer length field. The variable "X0" is a four byte variable and "X1" points to the first least significant byte of "X0" and "X2", the next significant and so on.
This way, the Write Buffer command will transfer the exact file size automatically.

CAUTION:
Depending on the vendor's implementation of the Write Buffer command, it may be damaging to the device to RESET or shut POWER OFF during the Write Buffer command. It is also suggested to set the command complete timeout value (see script line 9) to some value that is more than sufficient time to complete the Write Buffer command for the device you are working with.

Abort Limits
Hard Errors, ***154***
Soft Errors, ***154***
Appendix A - Image File Format, ***180***
Appendix B - Interpreting the Test Report, ***188***
Appendix C – Client Console Function, ***196***
Appendix D - Data Compare Description, ***197***
Appendix E - Command Timeout Description, ***202***
Appendix F - On Error Gosub Description, ***203***
Appendix G - Soft/Hard Error Description, ***204***
Appendix H - Error Rate Calculations, ***205***
Appendix I - Large File Download Example, ***206***
ATAPI Commands, ***51***
Build ATAPI Packet, ***52***
Configure Port (ATAPI), ***52***
Dasp / Pdiag Timing, ***56***
Data Streaming Test, ***53***
Generic Access Test, ***53***
Open / Close Tray, ***55***
Test CD Access Time Suite, ***56***
Test CD Transfer Rate Suite, ***56***
Binary Editor, ***44***
Buffer Selection, ***46***
Loading from Client Buffer, ***46***
Opening a File, ***45***
Other Features, ***46***
Printing the Editor Contents, ***46***
Saving Editor Contents to a File, ***45***
Saving to a Client Buffer, ***46***
Client Options, ***14***
Clear All Button, ***15***
Client Properties Table, ***15***
Description, ***15***
Maximum Ports, ***15***
Number of Columns, ***15***
Rack Number, ***15***
Commands by Group, ***77***
ATAPI Commands, ***51***
Command Terminators, ***118***
IDE Commands, ***64***
Loop & Utility Commands, ***56***
Oven Control Commands, ***63***
Program Control Commands, ***77***
Read Commands, ***91***
SCSI CCS Command Sub-Group, ***107***
SCSI Commands, ***98***
Seek Commands, ***124***
Set Parameter Commands, ***118***
Test Macro Commands, ***128***

Voltage Margin Commands, ***133***
Write Commands, ***95***
Database Output, ***47***
Legacy Files, ***47***
Directories, ***17***
Bootup Image Files, ***18***
Export Files, ***18***
Image Files, ***18***
Oven Scripts, ***18***
Report Files, ***18***
Test Scripts, ***18***
Glossary, ***178***
IDE Commands, ***64***
Configure Port, ***65***
Disable Drive Retries, ***66***
Download Microcode, ***67***
Initialize Drive Parameters, ***67***
Initiate Diagnostics, ***67***
Read Register, ***68***
Read Task File Regs, ***69***
Reset Drive, ***69***
Select ATA, ***70***
Set Data Mode, ***70***
Set Features Register, ***71***
Set Host DMA Mode, ***71***
Set Mem I/O Timing, ***71***
Set Multiple Mode, ***72***
Set Sector Overlay, ***73***
Set Slave Mode, ***73***
Set Xfer Length, ***73***
Vendor Unique, ***74***, ***75***
Write Register, ***76***
Installation, ***11***
Orion Host Software, ***11***, ***12***
Remote Boot Software, ***11***
Introduction, ***9***
Orion Host Software Description, ***9***
Product Features, ***9***
Loop & Utility Commands, ***56***
Chain to Program, ***57***
Clear Defect Buffer, ***57***
End Loop #1, ***57***
End Loop #2, ***58***
End Loop #3, ***58***
End Timed loop, ***58***
Force Test Failure, ***59***
LED Green On, ***59***
LED Red On, ***59***
Print Test Summary, ***60***
Return from Chained Program, ***60***

Start Loop #1, ***60***
Start Loop #2, ***61***
Start Loop #3, ***61***
Timed Loop, ***62***
Turn Off Both LEDs, ***59***
Update Image File, ***62***
Wait Delay, ***63***
Operation, ***25***
Pausing and Resuming a Test, ***28***
Port Icon Definitions, ***25***
Resetting Ports, ***27***
Starting a Test, ***27***
Stopping a Test, ***28***
Orion Host GUI, ***29***
Binary Editor Page, ***38***
Oven Status Page, ***37***
Performance Page, ***35***
Power Display Page, ***36***
Script Commands Page, ***29***
Status Page, ***34***
System Page, ***39***
Toolbar Buttons, ***40***
Orion Host Software Operating Instructions, ***12***
Bootup Options, ***25***
Client Options, ***14***
Directories, ***17***
Output Options, ***18***
Oven Options, ***21***
Preferences, ***24***
Server Options, ***13***
System Setup, ***12***
Orion Script Language, ***48***, ***51***
Abort Limits, ***154***
Commands by Group, ***51***
Using the Script Editor, ***48***
Variables, ***137***
Output Options, ***18***
Image File Update, ***20***
Interface Report Template Filename, ***20***
Report Filename, ***20***
Serial Number, ***19***
Oven Control, ***168***
Oven Script Editor Operation, ***169***
Oven Synchronization, ***172***
Script Language Description, ***171***
Oven Control Commands, ***63***
Run Oven program, ***63***
Wait for Oven Step, ***64***
Wait for Oven Temperature, ***64***
Oven Options, ***21***

Communication Settings, ***21***
Oven Graphing, ***22***
Oven Log File, ***22***
Oven Master Settings, ***22***
Reset Oven Communications, ***23***
Temperature Range Controls, ***22***
Oven Script Editor Operation, ***169***
Adding New Commands, ***170***
Checking Program Syntax, ***170***
Creating a New Script, ***170***
Deleting Commands, ***170***
Opening an Existing Script File, ***169***
Replacing Commands, ***170***
Saving a Script File, ***169***
Oven Script Language Description, ***171***
Autostart, ***171***
Comments, ***171***
Events, ***171***
LoopA, LoopB, LoopC, ***171***
Ramp, ***171***
Soak, ***171***
Oven Synchronization, ***172***
Oven Master Mode, ***174***
Oven Slave Mode, ***172***
Preferences, ***24***
Client Console, ***25***
Enable Engineering Display, ***24***
Manual Serial # Input, ***24***
Overwrite Performance Data, ***24***
Password, ***24***
Power On at Test End, ***24***
Spawn Command After Script Complete, ***25***
Program Control Commands, ***77***
Add Defect to List, ***78***
Add to Variable, ***78***
Add Variables, ***78***
And Variables, ***78***
Call Subroutine, ***79***
Comments, ***79***
Compare Buffers, ***80***
Compare Variables, ***80***
Copy Buffer to Buffer, ***81***
Custom Spawn, ***81***
Divide Variables, ***82***
Download File, ***82***
Filename, ***82***
Goto, ***83***
If Then Next Instruction, ***83***
Incr Hard Errors, ***84***
Incr Soft Errors, ***84***

Label, ***85***
Load Abort Limits, ***85***
Load Var from Buffer, ***85***
Move Var to Var, ***86***
Multiply Variable, ***86***
On Error Gosub, ***86***
Or Variable, ***87***
Pop Variable, ***87***
Push Variable, ***87***
Return from On Error Gosub, ***88***
Return from Subroutine, ***87***
Select Buffer, ***88***
Set Random Variable, ***89***
Set Variable, ***89***
Store Variable into Buffer, ***89***
Subtract from Variable, ***90***
Subtract Var from Var, ***90***
Upload File, ***90***
Xor Variable, ***91***
Read Commands, ***91***
Data Compare, ***91***
Read Current Cylinder, 92
Read Current Track, ***92***
Read Incr Forward, ***92***
Read Incr Reverse, ***93***
Read Oscillate, ***93***
Read Random Cylinder, ***94***
Read Random Track, ***94***
Read Sector, ***95***
Read Me First, ***8***
Orion Host Software Packing List, ***8***
System Requirements, ***8***
Report Generator Description, ***159***
Script Language Description, ***159***
Script Commands Page, ***29***
Chain History Page, ***33***
Command History Page, ***30***
Errors Page, ***32***
Report Output Page, ***30***
Variables Page, ***31***
Script Language Description, ***51***
Command List, ***160***
Keyword List, ***166***
SCSI CCS Command Sub-Group, ***107***
Change Definition, ***110***
Compare, ***110***
Copy, ***110***
Copy & Verify, ***110***
Edit Vendor Unique Command, ***111***
Erase, ***111***

Format Unit, ***111***
Inquiry, ***111***
Lock/Unlock Cache, ***111***
Log Select, ***112***
Log Sense, ***112***
Mode Select (10), ***112***
Mode Select (6), ***112***
Mode Sense (10), ***112***
Mode Sense (6), ***112***
Read (6), ***112***
Read Block limits, ***113***
Read Buffer, ***113***
Read Capacity, ***113***
Read Defect list, ***113***
Read Extended (10), ***113***
Read Long, ***113***
Read Reverse, ***114***
Read Verify, ***114***
Reassign Blocks, ***114***
Receive Diagnostics, ***114***
Recover Buffer Data, **114**
Release Unit, ***114***
Request Sense, ***114***
Reserve Unit, ***115***
Rezero Unit, ***115***
Search Equal, ***115***
Search High, ***115***
Search Low, ***115***
Seek Extended (10), ***116***
Seek Low, ***115***
Send Diagnostics, ***116***
Set Limits, ***116***
Start/Stop Unit, ***116***
Synchronize Cache, ***116***
Test Unit Ready, ***116***
Write (6), ***117***
Write and Verify, ***117***
Write Buffer, ***117***
Write Extended, ***117***
Write Long, ***117***
Write Same, ***117***
SCSI Commands, ***98***
1 - 8 Byte Pattern, ***102***
Build SCSI CDB, ***99***
Configure port, ***99***
Continue on Error, ***100***
Convert Defect List, ***100***
End Que, ***100***
If Cmd T.O. Gosub, ***99***
Load Zone Table, ***101***

Parity Control, ***101***
Read Defect List from Drive, ***101***
Reassign Blocks Macro, ***102***
Reset Drive, ***102***
Set #LBA to Xfer, ***105***
Set Auto Error Check, ***102***
Set Ident Message, ***103***
Set Initiator ID, ***103***
Set LUN, ***103***
Set SCSI Mode, ***104***
Set Target ID, ***106***
Start Que, ***106***
Sync Op Mode, ***104***
Test Parity, ***106***
Test Sync Data Transfers, ***107***
Seek Commands, ***124***
Decrement Cylinder, ***124***
Decrement Head, ***125***
Increment Cylinder, ***125***
Increment Head, ***125***
Seek Full Stroke, ***127***
Seek Home (Recalibrate), ***125***
Seek Increment Forward, ***126***
Seek Increment Reverse, ***126***
Seek Oscillate, ***127***
Seek Random Cylinder, ***127***
Seek to Cylinder, ***126***
Select Head, ***127***
Set Seek Increment, ***128***
Server Options, ***13***
DLC Protocol, ***13***
NE2000 Network Settings, ***14***
Process Message, ***14***
Rack Physical Layout, ***13***
System ID, ***14***
Set Parameter Commands, ***118***
Max Cylinder, ***121***
Max Head, ***122***
Min Cylinder, ***121***
Min Head, ***121***
Set Cmd Timeout, ***119***
Set Data Pattern-12 byte, ***120***
Set Data Pattern-2 byte, ***119***
Set Format Type Code, ***120***
Set Model Number, ***122***
Set Random Data Pattern, ***122***
Set Retries, ***123***
Set Serial Number, ***123***
Set Verify, ***123***
Table of Contents, ***2***

Technical Support, ***177***
Test Macro Commands, ***128***
End Performance, ***128***
Profile Current, ***129***
Start Performance, ***130***
Test Average Seek, ***130***
Test Drive Spin-up Time, ***130***
Test IDE I/O, ***132***
Test Index Timing, ***132***
Test Oscillating Seek Time, ***132***
Test Seek Timing, ***133***
Toolbar Buttons
Advance Step Button, ***43***
Command Step Mode Button, ***42***
Editors Button, ***41***
Generate Report Button, ***43***
Help Button, ***43***
Macro Step Mode Button, ***42***
Next Rack button, ***42***
Options Button, ***41***
Oven Pause/Resume Button, ***42***
Previous Rack Button, ***41***
Start Oven Button, ***42***
Stop Oven Button, ***42***
Troubleshooting, ***176***
Server will not Communicate with Clients, ***176***
Using the Script Editor, ***48***
Creating a New Script, ***48***
Deleting Script Commands, ***49***
Editing Abort Limits, ***50***
Inserting New Script Commands, ***49***
Modifying Parameters, ***49***
Printing Script Programs, ***50***
Saving Script Programs, ***50***
Starting the Editor, ***48***
Utilities, ***44***
Binary Editor, ***45***
Oven Script Editor, ***44***
Test Script Editor, ***44***
Variables, ***137***
Variable Details, ***140***
Voltage Margin Commands, ***133***
Drive Power Off, ***134***
Drive Power On, ***134***
Set Ch0 Volts, ***134***
Set Ch1 Volts, ***135***
Set Power Slew Rate, ***135***
Write Commands, ***95***
Oscillating Write, ***96***
Write Current Cylinder, ***96***

Write Current Track, ***95***
Write Incr Forward, ***96***
Write Incr Reverse, ***97***
Write Random Cylinder, ***97***
Write Random Track, ***97***
Write Sector, ***98***